InterSystems
DACH SYMPOSIUM 2023

TECHNOLOGY LAB

Moderne Webanwendungen mit InterSystems IRIS

Back to the future: Journeying Through Backend Technologies

1. Swagger Editor und Backend

Philipp Bonin | InterSystems

Schritte

Spezifizieren

Tool: Swagger Editor

Swagger Editor
Supported by SMARTBEAR

File ▾ Edit ▾ Generate Server ▾ Generate Client ▾ About ▾ Try our new Edito

```yaml
swagger: '2.0'
info:
  description: An API for Library Management using InterSystems IRIS
  version: 1.0.0
  title: Library Book
  license:
    name: Apache 2.0
    url: http://www.apache.org/licenses/LICENSE-2.0.html
schemes:
  - https
paths:
  /findabook/{bookid}:
    get:
      description: Find a book
      operationId: FindingBook
      produces:
        - application/json
      parameters:
        - name: bookid
          in: path
          type: string
          required: true
      responses:
        '200':
          description: Success
        '500':
          description: Server error
  /addfriend:
    post:
      description: Add a new friend
      operationId: AddingFriend
      produces:
        - application/json
      responses:
        '200':
          description: Success
        '500':
          description: Server error
  /addbook:
    post:
      description: Add a new book
      operationId: AddingBook
      produces:
        - application/json
      parameters:
        - name: title
          in: query
          type: string
```

Library Book 1.0.0 OAS 2.0

An API for Library Management using InterSystems IRIS

Apache 2.0

Schemes

HTTPS

default

GET /findabook/{bookid}

POST /addfriend

Add a new friend

Parameters Try it out

No parameters

Responses Response content type application/json

Swagger Editor
Supported by SMARTBEAR

File ▾ Edit ▾ Generate Server ▾ Generate Client ▾ About ▾ Try our new Edito

```yaml
swagger: '2.0'
info:
  description: An API for Library Management using InterSystems IRIS
  version: 1.0.0
  title: Library Book
  license:
    name: Apache 2.0
    url: http://www.apache.org/licenses/LICENSE-2.0.html
schemes:
  - https
paths:
  /findabook/{bookid}:
    get:
      description: Find a book
      operationId: FindingBook
      produces:
        - application/json
      parameters:
        - name: bookid
          in: path
          type: string
          required: true
      responses:
        '200':
          description: Success
        '500':
          description: Server error
  /addfriend:
    post:
      description: Add a new friend
      operationId: AddingFriend
      produces:
        - application/json
      responses:
        '200':
          description: Success
        '500':
          description: Server error
  /addbook:
    post:
      description: Add a new book
      operationId: AddingBook
      produces:
        - application/json
      parameters:
        - name: title
          in: query
          type: string
```

Library Book 1.0.0 OAS 2.0

An API for Library Management using InterSystems IRIS

Apache 2.0

Schemes

HTTPS

default

GET /findabook/{bookid}

POST /addfriend

Add a new friend

Parameters Try it out

No parameters

Responses Response content type application/json

```
    name: Apache 2.0
    url: http://www.apache.org/licenses/LICENSE-2.0.html
schemes:
  - https
paths:
  /findabook/{bookid}:
    get:
      description: Find a book
      operationId: FindingBook
      produces:
        - application/json
      parameters:
        - name: bookid
          in: path
          type: string
          required: true
      responses:
        '200':
          description: Success
        '500':
          description: Server error
  /addfriend:
    post:
      description: Add a new friend
      operationId: AddingFriend
      produces:
        - application/json
      responses:
        '200':
          description: Success
```

An API for Lib

Apache 2.0

Schemes

HTTPS

defaul

GET

POST

Add a new

```
    name: Apache 2.0
    url: http://www.apache.org/licenses/LICENSE-2.0.html
schemes:
  - https
paths:
  /findabook/{bookid}:
    get:
      description: Find a book
      operationId: FindingBook
      produces:
        - application/json
      parameters:
        - name: bookid
          in: path
          type: string
          required: true
      responses:
        '200':
          description: Success
        '500':
          description: Server error
  /addfriend:
    post:
      description: Add a new friend
      operationId: AddingFriend
      produces:
        - application/json
      responses:
        '200':
          description: Success
```

An API for Lib

Apache 2.0

Schemes

HTTPS

defaul

GET

POST

Add a new

```
    name: Apache 2.0
    url: http://www.apache.org/licenses/LICENSE-2.0.html
schemes:
  - https
paths:
  /findabook/{bookid}:
    get:
      description: Find a book
      operationId: FindingBook
      produces:
        - application/json
      parameters:
        - name: bookid
          in: path
          type: string
          required: true
      responses:
        '200':
          description: Success
        '500':
          description: Server error
  /addfriend:
    post:
      description: Add a new friend
      operationId: AddingFriend
      produces:
        - application/json
      responses:
        '200':
          description: Success
```

An API for Lib

Apache 2.0

Schemes

HTTPS

defaul

GET

POST

Add a new

```
    name: Apache 2.0
    url: http://www.apache.org/licenses/LICENSE-2.0.html
schemes:
  - https
paths:
  /findabook/{bookid}:
    get:
      description: Find a book
      operationId: FindingBook
      produces:
        - application/json
      parameters:
        - name: bookid
          in: path
          type: string
          required: true
      responses:
        '200':
          description: Success
        '500':
          description: Server error
  /addfriend:
    post:
      description: Add a new friend
      operationId: AddingFriend
      produces:
        - application/json
      responses:
        '200':
          description: Success
```

An API for Lib

Apache 2.0

Schemes

HTTPS

defaul

GET

POST

Add a new

```
    name: Apache 2.0
    url: http://www.apache.org/licenses/LICENSE-2.0.html
schemes:
  - https
paths:
  /findabook/{bookid}:
    get:
      description: Find a book
      operationId: FindingBook
      produces:
        - application/json
      parameters:
        - name: bookid
          in: path
          type: string
          required: true
      responses:
        '200':
          description: Success
        '500':
          description: Server error
  /addfriend:
    post:
      description: Add a new friend
      operationId: AddingFriend
      produces:
        - application/json
      responses:
        '200':
          description: Success
```

An API for Lib

Apache 2.0

Schemes

HTTPS

defaul

GET

POST

Add a new

```
    name: Apache 2.0
    url: http://www.apache.org/licenses/LICENSE-2.0.html
schemes:
  - https
paths:
  /findabook/{bookid}:
    get:
      description: Find a book
      operationId: FindingBook
      produces:
        - application/json
      parameters:
        - name: bookid
          in: path
          type: string
          required: true
      responses:
        '200':
          description: Success
        '500':
          description: Server error
  /addfriend:
    post:
      description: Add a new friend
      operationId: AddingFriend
      produces:
        - application/json
      responses:
        '200':
          description: Success
```

An API for Lib

Apache 2.0

Schemes

HTTPS

defaul

GET

POST

Add a new

```
    name: Apache 2.0
    url: http://www.apache.org/licenses/LICENSE-2.0.html
schemes:
  - https
paths:
  /findabook/{bookid}:
    get:
      description: Find a book
      operationId: FindingBook
      produces:
        - application/json
      parameters:
        - name: bookid
          in: path
          type: string
          required: true
      responses:
        '200':
          description: Success
        '500':
          description: Server error
  /addfriend:
    post:
      description: Add a new friend
      operationId: AddingFriend
      produces:
        - application/json
      responses:
        '200':
          description: Success
```

An API for Lib

Apache 2.0

Schemes

HTTPS

defaul

GET

POST

Add a new

```
    name: Apache 2.0
    url: http://www.apache.org/licenses/LICENSE-2.0.html
schemes:
  - https
paths:
  /findabook/{bookid}:
    get:
      description: Find a book
      operationId: FindingBook
      produces:
        - application/json
      parameters:
        - name: bookid
          in: path
          type: string
          required: true
      responses:
        '200':
          description: Success
        '500':
          description: Server error
  /addfriend:
    post:
      description: Add a new friend
      operationId: AddingFriend
      produces:
        - application/json
      responses:
        '200':
          description: Success
```

An API for Lib

Apache 2.0

Schemes

HTTPS

defaul

GET

POST

Add a new

```
    name: Apache 2.0
    url: http://www.apache.org/licenses/LICENSE-2.0.html
schemes:
  - https
paths:
  /findabook/{bookid}:
    get:
      description: Find a book
      operationId: FindingBook
      produces:
        - application/json
      parameters:
        - name: bookid
          in: path
          type: string
          required: true
      responses:
        '200':
          description: Success
        '500':
          description: Server error
  /addfriend:
    post:
      description: Add a new friend
      operationId: AddingFriend
      produces:
        - application/json
      responses:
        '200':
          description: Success
```

An API for Lib

Apache 2.0

Schemes

HTTPS

defaul

GET

POST

Add a new

```
    name: Apache 2.0
    url: http://www.apache.org/licenses/LICENSE-2.0.html
schemes:
  - https
paths:
  /findabook/{bookid}:
    get:
      description: Find a book
      operationId: FindingBook
      produces:
        - application/json
      parameters:
        - name: bookid
          in: path
          type: string
          required: true
      responses:
        '200':
          description: Success
        '500':
          description: Server error
  /addfriend:
    post:
      description: Add a new friend
      operationId: AddingFriend
      produces:
        - application/json
      responses:
        '200':
          description: Success
```

An API for Lib

Apache 2.0

Schemes

HTTPS

defaul

GET

POST

Add a new

Swagger Editor
Supported by SMARTBEAR

File ▾ Edit ▾ Generate Server ▾ Generate Client ▾ About ▾ Try our new Edito

```yaml
swagger: '2.0'
info:
  description: An API for Library Management using InterSystems IRIS
  version: 1.0.0
  title: Library Book
  license:
    name: Apache 2.0
    url: http://www.apache.org/licenses/LICENSE-2.0.html
schemes:
  - https
paths:
  /findabook/{bookid}:
    get:
      description: Find a book
      operationId: FindingBook
      produces:
        - application/json
      parameters:
        - name: bookid
          in: path
          type: string
          required: true
      responses:
        '200':
          description: Success
        '500':
          description: Server error
  /addfriend:
    post:
      description: Add a new friend
      operationId: AddingFriend
      produces:
        - application/json
      responses:
        '200':
          description: Success
        '500':
          description: Server error
  /addbook:
    post:
      description: Add a new book
      operationId: AddingBook
      produces:
        - application/json
      parameters:
        - name: title
          in: query
          type: string
```

Library Book 1.0.0 OAS 2.0

An API for Library Management using InterSystems IRIS

Apache 2.0

Schemes

HTTPS

default

GET /findabook/{bookid}

POST /addfriend

Add a new friend

Parameters Try it out

No parameters

Responses Response content type application/json

Swagger Editor
Supported by SMARTBEAR

File ▾ Edit ▾ Generate Server ▾ Generate Client ▾ About ▾ Try our new Edito

```yaml
swagger: '2.0'
info:
  description: An API for Library Management using InterSystems IRIS
  version: 1.0.0
  title: Library Book
  license:
    name: Apache 2.0
    url: http://www.apache.org/licenses/LICENSE-2.0.html
schemes:
  - https
paths:
  /findabook/{bookid}:
    get:
      description: Find a book
      operationId: FindingBook
      produces:
        - application/json
      parameters:
        - name: bookid
          in: path
          type: string
          required: true
      responses:
        '200':
          description: Success
        '500':
          description: Server error
  /addfriend:
    post:
      description: Add a new friend
      operationId: AddingFriend
      produces:
        - application/json
      responses:
        '200':
          description: Success
        '500':
          description: Server error
  /addbook:
    post:
      description: Add a new book
      operationId: AddingBook
      produces:
        - application/json
      parameters:
        - name: title
          in: query
          type: string
```

Library Book 1.0.0 OAS 2.0

An API for Library Management using InterSystems IRIS

Apache 2.0

Schemes

HTTPS

default

GET /findabook/{bookid}

POST /addfriend

Add a new friend

Parameters Try it out

No parameters

Responses Response content type application/json

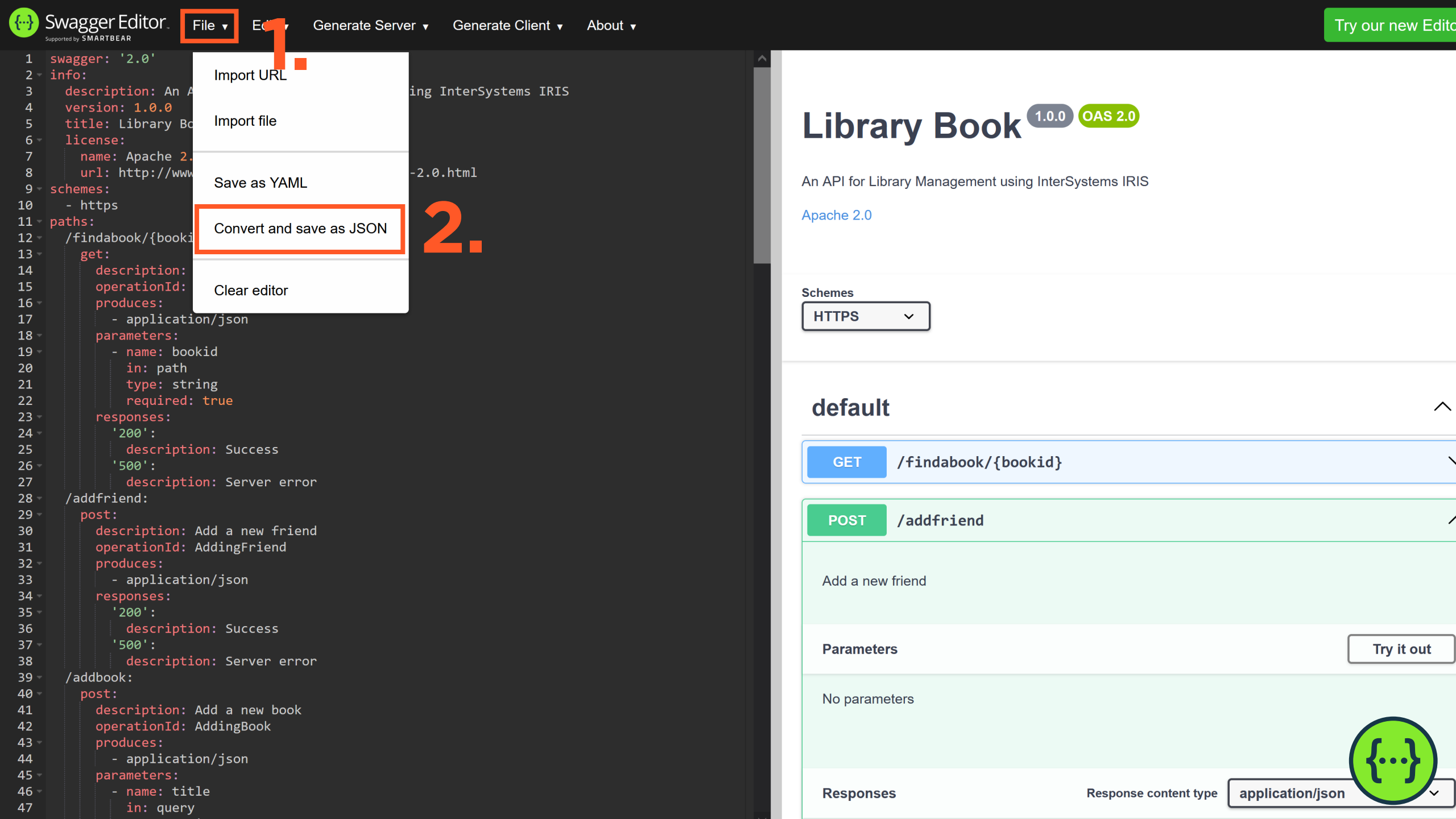
```
24          '200':
25            description: Success
26          '500':
27            description: Server error
28    /addfriend:
29      post:
30        description: Add a new friend
31        operationId: AddingFriend
32        produces:
33          - application/json
34        responses:
35          '200':
36            description: Success
37          '500':
38            description: Server error
39    /addbook:
40      post:
41        description: Add a new book
42        operationId: AddingBook
43        produces:
44          - application/json
45        parameters:
46          - name: title
47            in: query
48            type: string
```

```
24          '200':
25            description: Success
26          '500':
27            description: Server error
28    /addfriend:
29      post:
30        description: Add a new friend
31        operationId: AddingFriend
32        produces:
33          - application/json
34        responses:
35          '200':
36            description: Success
37          '500':
38            description: Server error
39    /addbook:
40      post:
41        description: Add a new book
42        operationId: AddingBook
43        produces:
44          - application/json
45        parameters:
46          - name: title
47            in: query
48            type: string
```

```
24          '200':
25            description: Success
26          '500':
27            description: Server error
28    /addfriend:
29      post:
30        description: Add a new friend
31        operationId: AddingFriend
32        produces:
33          - application/json
34        responses:
35          '200':
36            description: Success
37          '500':
38            description: Server error
39    /addbook:
40      post:
41        description: Add a new book
42        operationId: AddingBook
43        produces:
44          - application/json
45        parameters:
46          - name: title
47            in: query
48            type: string
```

```
24          '200':
25            description: Success
26          '500':
27            description: Server error
28    /addfriend:
29      post:
30        description: Add a new friend
31        operationId: AddingFriend
32        produces:
33          - application/json
34        responses:
35          '200':
36            description: Success
37          '500':
38            description: Server error
39    /addbook:
40      post:
41        description: Add a new book
42        operationId: AddingBook
43        produces:
44          - application/json
45        parameters:
46          - name: title
47            in: query
48            type: string
```

```
24          '200':
25            description: Success
26          '500':
27            description: Server error
28    /addfriend:
29      post:
30        description: Add a new friend
31        operationId: AddingFriend
32        produces:
33          - application/json
34        responses:
35          '200':
36            description: Success
37          '500':
38            description: Server error
39    /addbook:
40      post:
41        description: Add a new book
42        operationId: AddingBook
43        produces:
44          - application/json
45        parameters:
46          - name: title
47            in: query
48            type: string
```


Swagger Editor
Supported by SMARTBEAR
File
Edit
Generate Server
Generate Client
About
Try our new Edito
1.
Import URL
Import file
Save as YAML
Convert and save as JSON
2.
Clear editor
swagger: '2.0'
info:
description: An A ing InterSystems IRIS
version: 1.0.0
title: Library Bo
license:
name: Apache 2.
url: http://www -2.0.html
schemes:
- https
paths:
/findabook/{booki
get:
description:
operationId:
produces:
- application/json
parameters:
- name: bookid
in: path
type: string
required: true
responses:
'200':
description: Success
'500':
description: Server error
/addfriend:
post:
description: Add a new friend
operationId: AddingFriend
produces:
- application/json
responses:
'200':
description: Success
'500':
description: Server error
/addbook:
post:
description: Add a new book
operationId: AddingBook
produces:
- application/json
parameters:
- name: title
in: query
Library Book
1.0.0
OAS 2.0
An API for Library Management using InterSystems IRIS
Apache 2.0
Schemes
HTTPS
default
GET
/findabook/{bookid}
POST
/addfriend
Add a new friend
Parameters
Try it out
No parameters
Responses
Response content type
application/json

http://localhost:52773/api/mgmnt/v2/TRAINING/LibraryApi

POST http://localhost:52773/api/mgmnt/v2/TRAINING/LibraryApi

POST Request

Body • Pre-request Script Tests Settings

GraphQL JSON

```
{
    "swagger": "2.0",
    "info": {
        "description": "An API for Library Management using InterSystems IRIS",
        "version": "1.0.0",
        "title": "Library Book",
        "license": {
            "name": "Apache 2.0",
            "url": "http://www.apache.org/licenses/LICENSE-2.0.html"
        }
    },
    "schemes": ["https"],
    "paths": {
        "/addbook": {
            "post": {
                "description": "Add a new book",
                "operationId": "AddingBook",
                "produces": ["application/json"],
                "parameters": [
                    {
                        "name": "title",
                        "in": "query",
                        "type": "string"
```

Body Cookies Headers (9) Test Results

Status: 201 Created Time: 175 ms Size: 320 B

Pretty Raw Preview Visualize JSON

```
{
    "msg": "New application LibraryApi created"
}
```

http://localhost:52773/api/mgmnt/v2/TRAINING/LibraryApi

POST http://localhost:52773/api/mgmnt/v2/TRAINING/LibraryApi

Params Authorization Headers (10) Body Pre-request Script Tests Settings

none form-data x-www-form-urlencoded raw binary GraphQL JSON

IRIS Verbindung

```
{
    "swagger": "2.0",
    "info": {
        "description": "An API for Library Management using InterSystems IRIS",
        "version": "1.0.0",
        "title": "Library Book",
        "license": {
            "name": "Apache 2.0",
            "url": "http://www.apache.org/licenses/LICENSE-2.0.html"
        }
    },
    "schemes": ["https"],
    "paths": {
        "/addbook": {
            "post": {
                "description": "Add a new book",
                "operationId": "AddingBook",
                "produces": ["application/json"],
                "parameters": [
                    {
                        "name": "title",
                        "in": "query",
                        "type": "string"
                    }
```

Body Cookies Headers (9) Test Results

Status: 201 Created Time: 175 ms Size: 320 B

Pretty Raw Preview Visualize JSON

```
{
    "msg": "New application LibraryApi created"
}
```

http://localhost:52773/api/mgmnt/v2/TRAINING/LibraryApi

POST http://localhost:52773/api/mgmnt/v2/TRAINING/LibraryApi

Params Authorization Headers (10) Body Pre-request Script Tests Settings

API-Managment API

none form-data x-www-form-urlencoded raw binary GraphQL JSON

```
{
    "swagger": "2.0",
    "info": {
        "description": "An API for Library Management using InterSystems IRIS",
        "version": "1.0.0",
        "title": "Library Book",
        "license": {
            "name": "Apache 2.0",
            "url": "http://www.apache.org/licenses/LICENSE-2.0.html"
        }
    },
    "schemes": ["https"],
    "paths": {
        "/addbook": {
            "post": {
                "description": "Add a new book",
                "operationId": "AddingBook",
                "produces": ["application/json"],
                "parameters": [
                    {
                        "name": "title",
                        "in": "query",
                        "type": "string"
                    }
```

Body Cookies Headers (9) Test Results

Status: 201 Created Time: 175 ms Size: 320 B

Pretty Raw Preview Visualize JSON

```
{
    "msg": "New application LibraryApi created"
}
```

http://localhost:52773/api/mgmnt/v2/TRAINING/LibraryApi

POST http://localhost:52773/api/mgmnt/v2/TRAINING/LibraryApi

Params Authorization Headers (1) Body Pre-request Script Tests Settings

Namespace

none form-data x-www-form-urlencoded raw binary GraphQL JSON

```
{
    "swagger": "2.0",
    "info": {
        "description": "An API for Library Management using InterSystems IRIS",
        "version": "1.0.0",
        "title": "Library Book",
        "license": {
            "name": "Apache 2.0",
            "url": "http://www.apache.org/licenses/LICENSE-2.0.html"
        }
    },
    "schemes": ["https"],
    "paths": {
        "/addbook": {
            "post": {
                "description": "Add a new book",
                "operationId": "AddingBook",
                "produces": ["application/json"],
                "parameters": [
                    {
                        "name": "title",
                        "in": "query",
                        "type": "string"
                    }
```

Body Cookies Headers (9) Test Results

Status: 201 Created Time: 175 ms Size: 320 B

Pretty Raw Preview Visualize JSON

```
{
    "msg": "New application LibraryApi created"
}
```

http://localhost:52773/api/mgmnt/v2/TRAINING/LibraryApi

POST http://localhost:52773/api/mgmnt/v2/TRAINING/LibraryApi

Params Authorization Headers (1) Body Pre-request Script Tests Settings

none form-data x-www-form-urlencoded raw binary GraphQL JSON

Name der Webapplikation

```
{
    "swagger": "2.0",
    "info": {
        "description": "An API for Library Management using InterSystems IRIS",
        "version": "1.0.0",
        "title": "Library Book",
        "license": {
            "name": "Apache 2.0",
            "url": "http://www.apache.org/licenses/LICENSE-2.0.html"
        }
    },
    "schemes": ["https"],
    "paths": {
        "/addbook": {
            "post": {
                "description": "Add a new book",
                "operationId": "AddingBook",
                "produces": ["application/json"],
                "parameters": [
                    {
                        "name": "title",
                        "in": "query",
                        "type": "string"
                    }
```

Body Cookies Headers (9) Test Results

Status: 201 Created Time: 175 ms Size: 320 B

Pretty Raw Preview Visualize JSON

```
{
    "msg": "New application LibraryApi created"
}
```

http://localhost:52773/api/mgmnt/v2/TRAINING/LibraryApi

POST http://localhost:52773/api/mgmnt/v2/TRAINING/LibraryApi

Params Authorization Headers (10) Body Pre-request Script Tests Settings

none form-data x-www-form-urlencoded raw binary GraphQL JSON

Authentifizieren

```
{
    "swagger": "2.0",
    "info": {
        "description": "An API for Library Management using InterSystems IRIS",
        "version": "1.0.0",
        "title": "Library Book",
        "license": {
            "name": "Apache 2.0",
            "url": "http://www.apache.org/licenses/LICENSE-2.0.html"
        }
    },
    "schemes": ["https"],
    "paths": {
        "/addbook": {
            "post": {
                "description": "Add a new book",
                "operationId": "AddingBook",
                "produces": ["application/json"],
                "parameters": [
                    {
                        "name": "title",
                        "in": "query",
                        "type": "string"
                    }
```

Body Cookies Headers (9) Test Results

Status: 201 Created Time: 175 ms Size: 320 B

Pretty Raw Preview Visualize JSON

```
{
    "msg": "New application LibraryApi created"
}
```

http://localhost:52773/api/mgmnt/v2/TRAINING/LibraryApi

POST http://localhost:52773/api/mgmnt/v2/TRAINING/LibraryApi

Params Authorization Headers (10) Body Pre-request Script Tests Settings

none form-data x-www-form-urlencoded raw binary GraphQL JSON

```
{
    "swagger": "2.0",
    "info": {
        "description": "An API for Library Management using InterSystems IRIS",
        "version": "1.0.0",
        "title": "Library Book",
        "license": {
            "name": "Apache 2.0",
            "url": "http://www.apache.org/licenses/LICENSE-2.0.html"
        }
    },
    "schemes": ["https"],
    "paths": {
        "/addbook": {
            "post": {
                "description": "Add a new book",
                "operationId": "AddingBook",
                "produces": ["application/json"],
                "parameters": [
                    {
                        "name": "title",
                        "in": "query",
                        "type": "string"
```

Spezifikation

Body Cookies Headers (9) Test Results

Status: 201 Created Time: 175 ms Size: 320 B

Pretty Raw Preview Visualize JSON

```
{
    "msg": "New application LibraryApi created"
}
```

http://localhost:52773/api/mgmnt/v2/TRAINING/LibraryApi

POST http://localhost:52773/api/mgmnt/v2/TRAINING/LibraryApi

Params Authorization Headers (10) Body Pre-request Script Tests Settings

none form-data x-www-form-urlencoded raw binary GraphQL JSON

```
{
    "swagger": "2.0",
    "info": {
        "description": "An API for Library Management using InterSystems IRIS",
        "version": "1.0.0",
        "title": "Library Book",
        "license": {
            "name": "Apache 2.0",
            "url": "http://www.apache.org/licenses/LICENSE-2.0.html"
        }
    },
    "schemes": ["https"],
    "paths": {
        "/addbook": {
            "post": {
                "description": "Add a new book",
                "operationId": "AddingBook",
                "produces": ["application/json"],
                "parameters": [
                    {
                        "name": "title",
                        "in": "query",
                        "type": "string"
```

Body Cookies Header esults Status: 201 Created Time: 175 ms Size: 320 B

Pretty Raw Preview Visualize JSON

```
{
    "msg": "New application LibraryApi created"
}
```

Webanwendung erstellen

Willkommen, _SYSTEM

Ansicht:

- Startseite
- Analytics
- Interoperability
- Systembetrieb
- System-Explorer
- Systemadministration

Konfiguration	Benutzer	Webanwendungen
Sicherheit	Rollen	Doc DB-Anwendungen
Lizenzierung	Ressourcen	Privilegierte Routinen-Applikationen
ENCRYPTION	Dienste	Client-Applikationen
	Applikationen	
	SSL/TLS-Konfigurationen	
	X.509-Informationen	
	OAuth 2.0	
	Verwaltete Dateiübertragungsverbindun	
	Systemsicherheit	
	Auditdurchführung	
	Sicherheitsberater	
	Mobiltelefon	
	Public-Key-Infrastruktur	

WEBANWENDUNGEN

Web-Anwendungsdefinitionen anzeigen, hinzufügen oder bearbeiten.

Los

Zu Favoriten hinzufügen

Systemressource(n)
%Admin_Secure

Benutzerdefinierte Ressource
-

Zuweisen

Webanwendung erstellen

System > Sicherheitsmanagement > Webanwendungen > Webanwendung bearbeiten - (Sicherheitseinstellungen)

Webanwendung bearbeiten

Speichern | Abbrechen

Definition für Webanwendung /LibraryApi bearbeiten:

Allgemein | Applikationsrollen | Partnerrollen

Name /LibraryApi
Erforderlich. (z. B. /csp/appname)

Beschreibung

NameSpace TRAINING **Standardanwendung für TRAINING:** /csp/training Namespace-Standardanwendung

Anwendung aktivieren

Aktivieren
- **REST**
  - **Verteilerklasse** LibraryApi.disp
    Erforderlich.
  - **Leeren Pfad umleiten**
  - **Use JWT Authentication**
- **CSP/ZEN**
  - Analyse | Eingehende Webdienste | Login-CSRF-Angriff verhindern

Sicherheitseinstellungen
- **Erforderliche Ressource** **Nach ID gruppieren**
- **Zulässige Authentifizierungsmethoden** Nicht authentifiziert | Kennwort | Anmelde-Cookie

Sitzungseinstellungen
- **Sitzungs-Timeout** 900 Sekunden **Ereignisklasse** .cls
- **Cookie für Sitzung verwenden** Immer **Pfad zu Sitzungs-Cookie** /LibraryApi/ **Sitzungs-Cookie-Umfang** Strikt **Benutzer-Cookie-Geltungsbereich** Strikt

Webanwendung erstellen

System > Sicherheitsmanagement > Webanwendungen > Webanwendung bearbeiten - (Sicherheitseinstellungen)

Webanwendung bearbeiten

Speichern | Abbrechen

Definition für Webanwendung /LibraryApi bearbeiten:

Allgemein | Applikationsrollen | Partnerrollen

Name /LibraryApi
Erforderlich. (z. B. /csp/appname)

Beschreibung

NameSpace TRAINING — **Standardanwendung für TRAINING:** /csp/training — Namespace-Standardanwendung

Anwendung aktivieren

Aktivieren

- **REST**
  - **Verteilerklasse** LibraryApi.disp
    Erforderlich.
  - **Leeren Pfad umleiten**
  - **Use JWT Authentication**
- **CSP/ZEN**
  - Analyse — Eingehende Webdienste — Login-CSRF-Angriff verhindern

Sicherheitseinstellungen

Erforderliche Ressource — **Nach ID gruppieren**

Zulässige Authentifizierungsmethoden Nicht authentifiziert — Kennwort — Anmelde-Cookie

Sitzungseinstellungen

Sitzungs-Timeout 900 Sekunden — **Ereignisklasse** .cls

Cookie für Sitzung verwenden Immer — **Pfad zu Sitzungs-Cookie** /LibraryApi/ — **Sitzungs-Cookie-Umfang** Strikt — **Benutzer-Cookie-Geltungsbereich** Strikt

Webanwendung erstellen

System > Sicherheitsmanagement > Webanwendungen > Webanwendung bearbeiten - (Sicherheitseinstellungen)

Webanwendung bearbeiten

Speichern | Abbrechen

Definition für Webanwendung /LibraryApi bearbeiten:

Allgemein | Applikationsrollen | Partnerrollen

Name /LibraryApi
Erforderlich. (z. B. /csp/appname)

Beschreibung

NameSpace TRAINING **Standardanwendung für TRAINING:** /csp/training Namespace-Standardanwendung

Anwendung aktivieren

Aktivieren **REST**

Verteilerklasse LibraryApi.disp
Erforderlich.

Leeren Pfad umleiten

Use JWT Authentication

CSP/ZEN

Analyse Eingehende Webdienste Login-CSRF-Angriff verhindern

Sicherheitseinstellungen

Erforderliche Ressource **Nach ID gruppieren**

Zulässige Authentifizierungsmethoden Nicht authentifiziert Kennwort Anmelde-Cookie

Sitzungseinstellungen

Sitzungs-Timeout 900 Sekunden **Ereignisklasse** .cls

Cookie für Sitzung verwenden Immer **Pfad zu Sitzungs-Cookie** /LibraryApi/ **Sitzungs-Cookie-Umfang** Strikt **Benutzer-Cookie-Geltungsbereich** Strikt

Webanwendung erstellen

System > Sicherheitsmanagement > Webanwendungen > Webanwendung bearbeiten - (Sicherheitseinstellungen)

Webanwendung bearbeiten

Speichern | Abbrechen

Definition für Webanwendung /LibraryApi bearbeiten:

Allgemein | Applikationsrollen | Partnerrollen

Name /LibraryApi
Erforderlich. (z. B. /csp/appname)

Beschreibung

NameSpace TRAINING **Standardanwendung für TRAINING:** /csp/training Namespace-Standardanwendung

Anwendung aktivieren

Aktivieren
- **REST**
  - **Verteilerklasse** LibraryApi.disp
    Erforderlich.
  - **Leeren Pfad umleiten**
  - **Use JWT Authentication**
- **CSP/ZEN**
  - Analyse | Eingehende Webdienste | Login-CSRF-Angriff verhindern

Sicherheitseinstellungen
- **Erforderliche Ressource** **Nach ID gruppieren**
- **Zulässige Authentifizierungsmethoden** Nicht authentifiziert | Kennwort | Anmelde-Cookie

Sitzungseinstellungen
- **Sitzungs-Timeout** 900 Sekunden **Ereignisklasse** .cls
- **Cookie für Sitzung verwenden** Immer **Pfad zu Sitzungs-Cookie** /LibraryApi/ **Sitzungs-Cookie-Umfang** Strikt **Benutzer-Cookie-Geltungsbereich** Strikt

Webanwendung erstellen

System > Sicherheitsmanagement > Webanwendungen > Webanwendung bearbeiten - (Sicherheitseinstellungen)

Webanwendung bearbeiten

Speichern | Abbrechen

Definition für Webanwendung /LibraryApi bearbeiten:

Allgemein | Applikationsrollen | Partnerrollen

Name /LibraryApi
Erforderlich. (z. B. /csp/appname)

Beschreibung

NameSpace TRAINING **Standardanwendung für TRAINING:** /csp/training Namespace-Standardanwendung

Anwendung aktivieren

Aktivieren
- **REST**
  - **Verteilerklasse** LibraryApi.disp
    Erforderlich.
  - **Leeren Pfad umleiten**
  - **Use JWT Authentication**
- **CSP/ZEN**
  - Analyse Eingehende Webdienste Login-CSRF-Angriff verhindern

Sicherheitseinstellungen
- **Erforderliche Ressource** **Nach ID gruppieren**
- **Zulässige Authentifizierungsmethoden** Nicht authentifiziert Kennwort Anmelde-Cookie

Sitzungseinstellungen
- **Sitzungs-Timeout** 900 Sekunden **Ereignisklasse** .cls
- **Cookie für Sitzung verwenden** Immer **Pfad zu Sitzungs-Cookie** /LibraryApi/ **Sitzungs-Cookie-Umfang** Strikt **Benutzer-Cookie-Geltungsbereich** Strikt

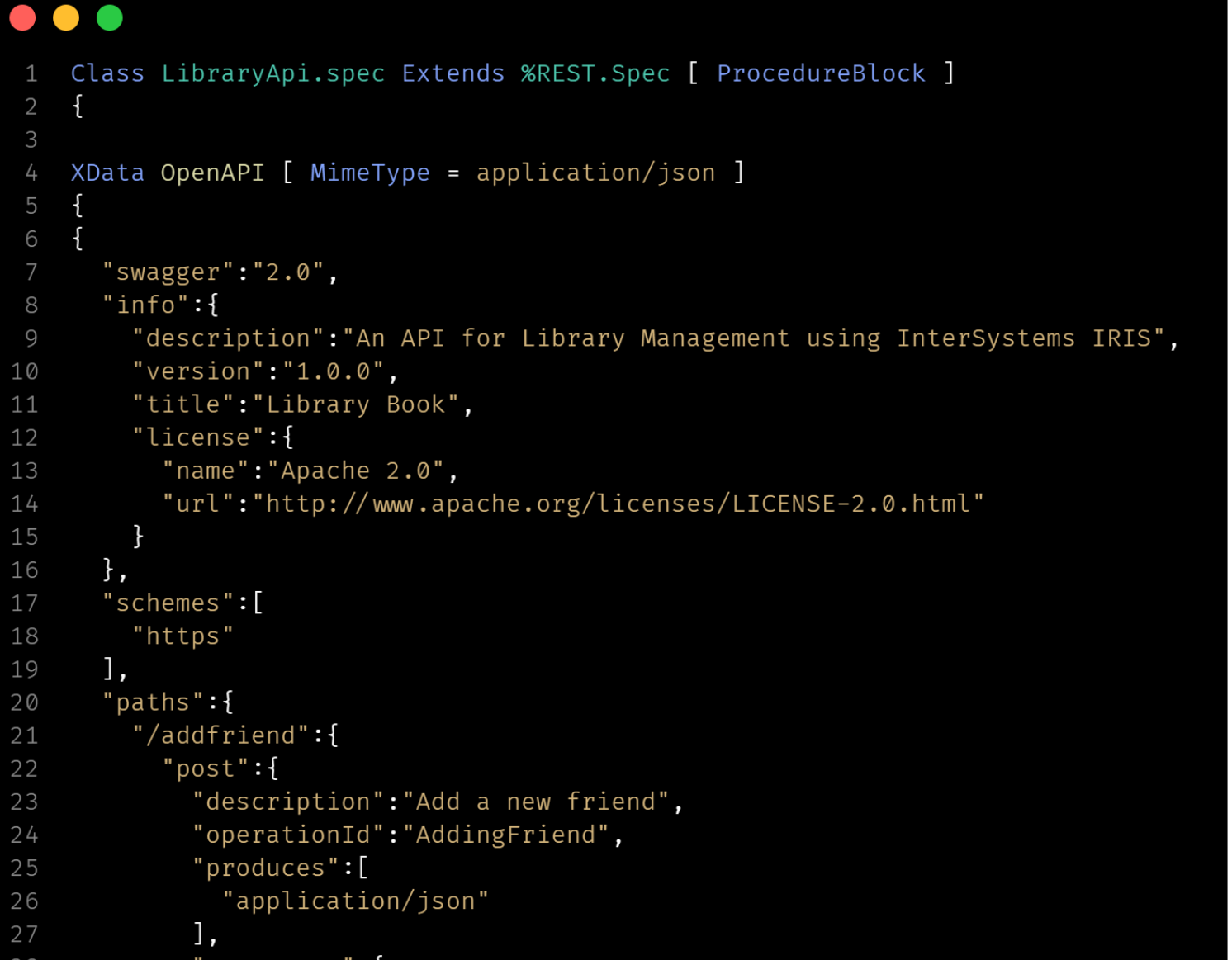
Bonus: Swagger Editor for InterSystems

InterSystems®
DACH SYMPOSIUM 2023

Implementieren

Tool: Visual Studio Code

LibraryApi.spec

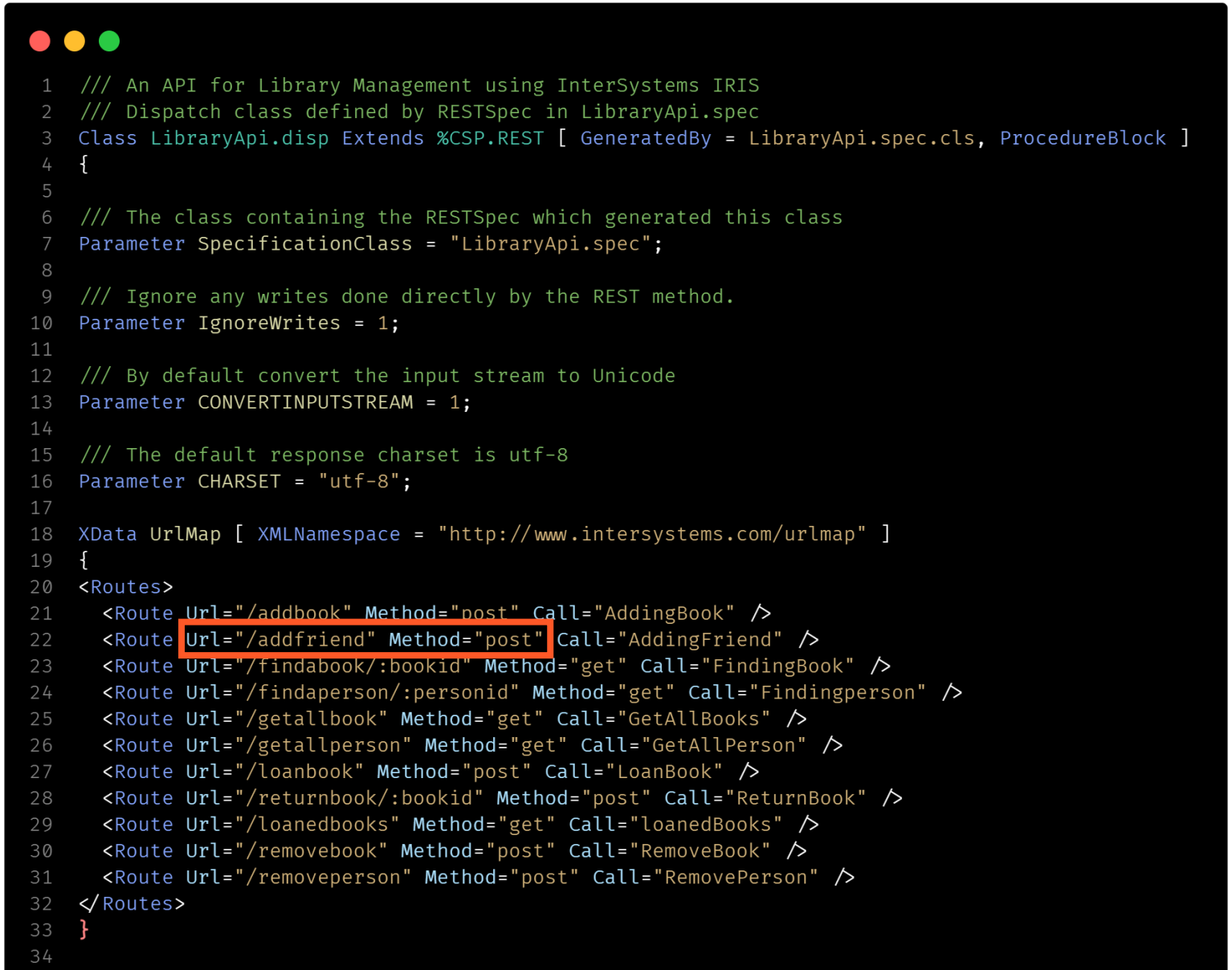


```
Class LibraryApi.spec Extends %REST.Spec [ ProcedureBlock ]
{

XData OpenAPI [ MimeType = application/json ]
{
{
  "swagger":"2.0",
  "info":{
    "description":"An API for Library Management using InterSystems IRIS",
    "version":"1.0.0",
    "title":"Library Book",
    "license":{
      "name":"Apache 2.0",
      "url":"http://www.apache.org/licenses/LICENSE-2.0.html"
    }
  },
  "schemes":[
    "https"
  ],
  "paths":{
    "/addfriend":{
      "post":{
        "description":"Add a new friend",
        "operationId":"AddingFriend",
        "produces":[
          "application/json"
        ],
```

LibraryApi.disp

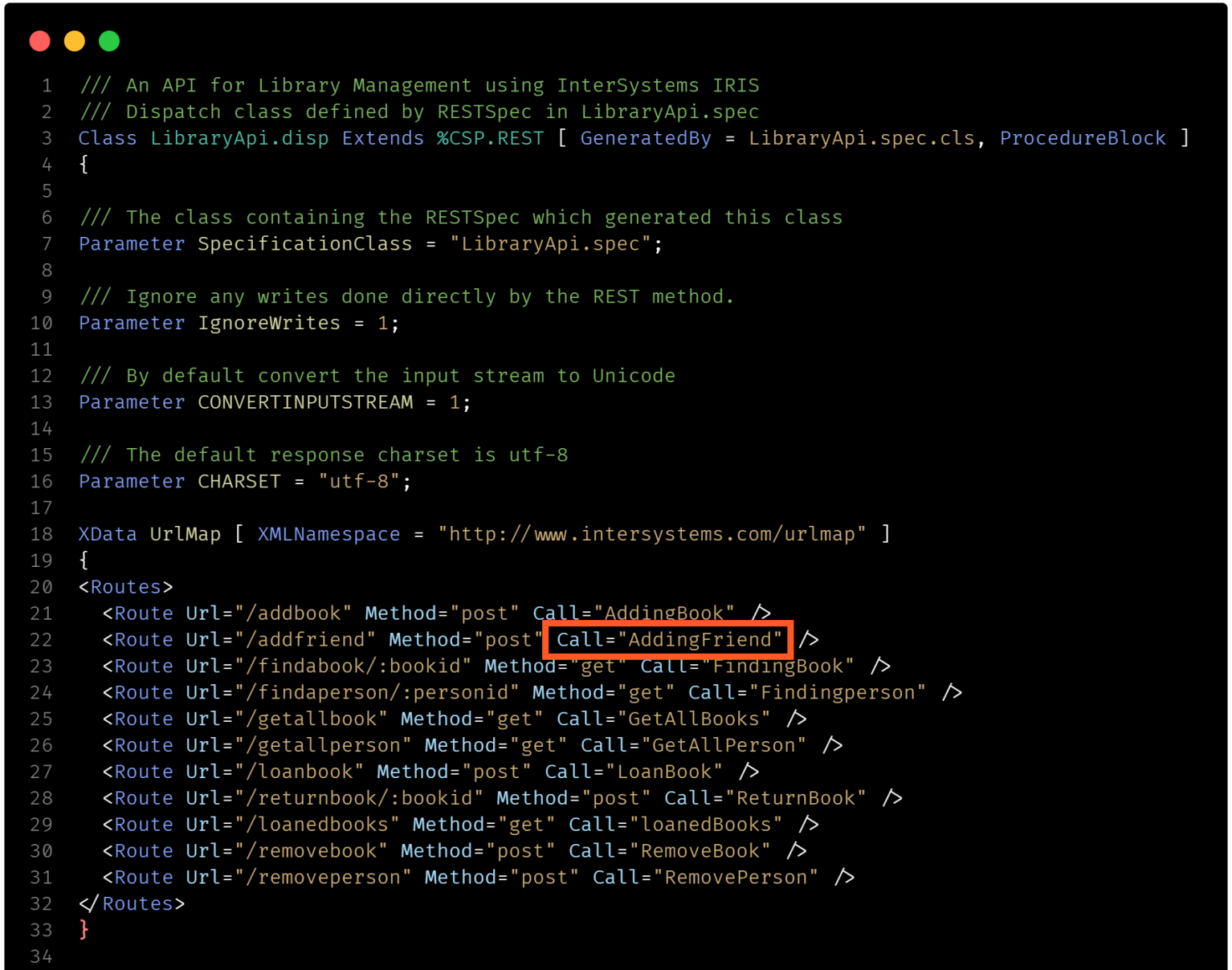
```
/// An API for Library Management using InterSystems IRIS
/// Dispatch class defined by RESTSpec in LibraryApi.spec
Class LibraryApi.disp Extends %CSP.REST [ GeneratedBy = LibraryApi.spec.cls, ProcedureBlock ]
{

/// The class containing the RESTSpec which generated this class
Parameter SpecificationClass = "LibraryApi.spec";

/// Ignore any writes done directly by the REST method.
Parameter IgnoreWrites = 1;

/// By default convert the input stream to Unicode
Parameter CONVERTINPUTSTREAM = 1;

/// The default response charset is utf-8
Parameter CHARSET = "utf-8";

XData UrlMap [ XMLNamespace = "http://www.intersystems.com/urlmap" ]
{
<Routes>
  <Route Url="/addbook" Method="post" Call="AddingBook" />
  <Route Url="/addfriend" Method="post" Call="AddingFriend" />
  <Route Url="/findabook/:bookid" Method="get" Call="FindingBook" />
  <Route Url="/findaperson/:personid" Method="get" Call="Findingperson" />
  <Route Url="/getallbook" Method="get" Call="GetAllBooks" />
  <Route Url="/getallperson" Method="get" Call="GetAllPerson" />
  <Route Url="/loanbook" Method="post" Call="LoanBook" />
  <Route Url="/returnbook/:bookid" Method="post" Call="ReturnBook" />
  <Route Url="/loanedbooks" Method="get" Call="loanedBooks" />
  <Route Url="/removebook" Method="post" Call="RemoveBook" />
  <Route Url="/removeperson" Method="post" Call="RemovePerson" />
</Routes>
}

```

LibraryApi.disp



```
/// An API for Library Management using InterSystems IRIS
/// Dispatch class defined by RESTSpec in LibraryApi.spec
Class LibraryApi.disp Extends %CSP.REST [ GeneratedBy = LibraryApi.spec.cls, ProcedureBlock ]
{

/// The class containing the RESTSpec which generated this class
Parameter SpecificationClass = "LibraryApi.spec";

/// Ignore any writes done directly by the REST method.
Parameter IgnoreWrites = 1;

/// By default convert the input stream to Unicode
Parameter CONVERTINPUTSTREAM = 1;

/// The default response charset is utf-8
Parameter CHARSET = "utf-8";

XData UrlMap [ XMLNamespace = "http://www.intersystems.com/urlmap" ]
{
<Routes>
  <Route Url="/addbook" Method="post" Call="AddingBook" />
  <Route Url="/addfriend" Method="post" Call="AddingFriend" />
  <Route Url="/findabook/:bookid" Method="get" Call="FindingBook" />
  <Route Url="/findaperson/:personid" Method="get" Call="Findingperson" />
  <Route Url="/getallbook" Method="get" Call="GetAllBooks" />
  <Route Url="/getallperson" Method="get" Call="GetAllPerson" />
  <Route Url="/loanbook" Method="post" Call="LoanBook" />
  <Route Url="/returnbook/:bookid" Method="post" Call="ReturnBook" />
  <Route Url="/loanedbooks" Method="get" Call="loanedBooks" />
  <Route Url="/removebook" Method="post" Call="RemoveBook" />
  <Route Url="/removeperson" Method="post" Call="RemovePerson" />
</Routes>
}

```

LibraryApi.disp



```
/// An API for Library Management using InterSystems IRIS
/// Dispatch class defined by RESTSpec in LibraryApi.spec
Class LibraryApi.disp Extends %CSP.REST [ GeneratedBy = LibraryApi.spec.cls, ProcedureBlock ]
{

/// The class containing the RESTSpec which generated this class
Parameter SpecificationClass = "LibraryApi.spec";

/// Ignore any writes done directly by the REST method.
Parameter IgnoreWrites = 1;

/// By default convert the input stream to Unicode
Parameter CONVERTINPUTSTREAM = 1;

/// The default response charset is utf-8
Parameter CHARSET = "utf-8";

XData UrlMap [ XMLNamespace = "http://www.intersystems.com/urlmap" ]
{
<Routes>
  <Route Url="/addbook" Method="post" Call="AddingBook" />
  <Route Url="/addfriend" Method="post" Call="AddingFriend" />
  <Route Url="/findabook/:bookid" Method="get" Call="FindingBook" />
  <Route Url="/findaperson/:personid" Method="get" Call="Findingperson" />
  <Route Url="/getallbook" Method="get" Call="GetAllBooks" />
  <Route Url="/getallperson" Method="get" Call="GetAllPerson" />
  <Route Url="/loanbook" Method="post" Call="LoanBook" />
  <Route Url="/returnbook/:bookid" Method="post" Call="ReturnBook" />
  <Route Url="/loanedbooks" Method="get" Call="loanedBooks" />
  <Route Url="/removebook" Method="post" Call="RemoveBook" />
  <Route Url="/removeperson" Method="post" Call="RemovePerson" />
</Routes>
}

```

LibraryApi.impl

```
1  /// An API for Library Management using InterSystems IRIS<br/>
2  /// Business logic class defined by OpenAPI in LibraryApi.spec<br/>
3  /// Created Nov 22, 2023 11:07:06
4  Class LibraryApi.impl Extends %REST.Impl [ ProcedureBlock ]
5  {
6
7  /// If ExposeServerExceptions is true, then details of internal errors will be exposed.
8  Parameter ExposeServerExceptions = 0;
9
10  ...
11
12  }
```

Tipp: Debugging made easy

```
1  /// An API for Library Management using InterS
2  /// Business logic class defined by OpenAPI in
3  /// Created Nov 22, 2023 11:07:06
4  Class LibraryApi.impl Extends %REST.Impl [ Pro
5  {
6
7  /// If ExposeServerExceptions is true, then de
8  Parameter ExposeServerExceptions = 1;
9
10  ...
11
12  }
```


NICHT IN PRODUKTION!!!!

GET /findabook/{bookid}

```
1  /// Find a book<br/>
2  /// The method arguments hold values for:<br/>
3  ///     bookid<br/>
4  ClassMethod FindingBook(bookid As %String) As %DynamicObject
5  {
6
7
8
9
10
11
12
13
14
15
16 }
```

```
    name: Apache 2.0
    url: http://www.apache.org/licenses/LICENSE-2.0.html
schemes:
  - https
paths:
  /findabook/{bookid}:
    get:
      description: Find a book
      operationId: FindingBook
      produces:
        - application/json
      parameters:
        - name: bookid
          in: path
          type: string
          required: true
      responses:
        '200':
          description: Success
        '500':
          description: Server error
  /addfriend:
    post:
      description: Add a new friend
      operationId: AddingFriend
      produces:
        - application/json
      responses:
        '200':
          description: Success
```

An API for Lib

Apache 2.0

Schemes

HTTPS

defaul

GET

POST

Add a new

GET /findabook/{bookid}

```
1  /// Find a book<br/>
2  /// The method arguments hold values for:<br/>
3  ///     bookid<br/>
4  ClassMethod FindingBook(bookid As %String) As %DynamicObject
5  {
6
7      Set result = ##class(Library.LibraryUtils).GetBook(bookid, .foundBook)
8
9
10
11
12
13
14
15
16 }
```

GET /findabook/{bookid}

```
1  /// Find a book<br/>
2  /// The method arguments hold values for:<br/>
3  ///     bookid<br/>
4  ClassMethod FindingBook(bookid As %String) As %DynamicObject
5  {
6
7      Set result = ##class(Library.LibraryUtils).GetBook(bookid, .foundBook)
8      If result = 0{
9           Do ..%SetStatusCode(500)
10          Return "something is wrong"
11     }
12
13
14
15
16 }
```

GET /findabook/{bookid}

```
1  /// Find a book<br/>
2  /// The method arguments hold values for:<br/>
3  ///     bookid<br/>
4  ClassMethod FindingBook(bookid As %String) As %DynamicObject
5  {
6
7      Set result = ##class(Library.LibraryUtils).GetBook(bookid, .foundBook)
8      If result = 0{
9           Do ..%SetStatusCode(500)
10          Return "something is wrong"
11     }
12     Do ..%SetStatusCode(200)
13
14
15
16 }
```

GET /findabook/{bookid}

```
1  /// Find a book<br/>
2  /// The method arguments hold values for:<br/>
3  ///     bookid<br/>
4  ClassMethod FindingBook(bookid As %String) As %DynamicObject
5  {
6
7      Set result = ##class(Library.LibraryUtils).GetBook(bookid, .foundBook)
8      If result = 0{
9           Do ..%SetStatusCode(500)
10          Return "something is wrong"
11     }
12     Do ..%SetStatusCode(200)
13
14     Set sc = foundBook.%JSONExportToString(.s)
15     Return s
16 }
```

POST /addfriend

```
1  /// Add a new friend
2  ClassMethod AddingFriend() As %DynamicObject
3  {
4
5
6
7
8
9
10
11
12
13
14
15
16
17 }
```

```
24          '200':
25            description: Success
26          '500':
27            description: Server error
28    /addfriend:
29      post:
30        description: Add a new friend
31        operationId: AddingFriend
32        produces:
33          - application/json
34        responses:
35          '200':
36            description: Success
37          '500':
38            description: Server error
39    /addbook:
40      post:
41        description: Add a new book
42        operationId: AddingBook
43        produces:
44          - application/json
45        parameters:
46          - name: title
47            in: query
48            type: string
```

POST /addfriend

```
/// Add a new friend
ClassMethod AddingFriend() As %DynamicObject
{
    Set body = ##class(%DynamicObject).%FromJSON(%request.Content)



}
```

POST /addfriend

```
/// Add a new friend
ClassMethod AddingFriend() As %DynamicObject
{
    Set body = ##class(%DynamicObject).%FromJSON(%request.Content)
    Set friendName = body.%Get("friendName")
    Set friendNumber = body.%Get("friendNumber")
    Set status =##class(Library.LibraryUtils).AddFriend(friendName,friendNumber)


}
```

POST /addfriend

```
/// Add a new friend
ClassMethod AddingFriend() As %DynamicObject
{
    Set body = ##class(%DynamicObject).%FromJSON(%request.Content)
    Set friendName = body.%Get("friendName")
    Set friendNumber = body.%Get("friendNumber")
    Set status =##class(Library.LibraryUtils).AddFriend(friendName,friendNumber)

    Set jsonresponse = ##class(%DynamicObject).%New()
    Do ..%SetStatusCode(200)



}
```

POST /addfriend

```
/// Add a new friend
ClassMethod AddingFriend() As %DynamicObject
{
    Set body = ##class(%DynamicObject).%FromJSON(%request.Content)
    Set friendName = body.%Get("friendName")
    Set friendNumber = body.%Get("friendNumber")
    Set status =##class(Library.LibraryUtils).AddFriend(friendName,friendNumber)

    Set jsonresponse = ##class(%DynamicObject).%New()
    Do ..%SetStatusCode(200)
    If status = 0{
        Do ..%SetStatusCode(500)
        Set status = 0
    }


}
```

POST /addfriend

```
/// Add a new friend
ClassMethod AddingFriend() As %DynamicObject
{
    Set body = ##class(%DynamicObject).%FromJSON(%request.Content)
    Set friendName = body.%Get("friendName")
    Set friendNumber = body.%Get("friendNumber")
    Set status =##class(Library.LibraryUtils).AddFriend(friendName,friendNumber)

    Set jsonresponse = ##class(%DynamicObject).%New()
    Do ..%SetStatusCode(200)
    If status = 0{
        Do ..%SetStatusCode(500)
        Set status = 0
    }
    Do jsonresponse.%Set("status",status)
    Return jsonresponse
}
```

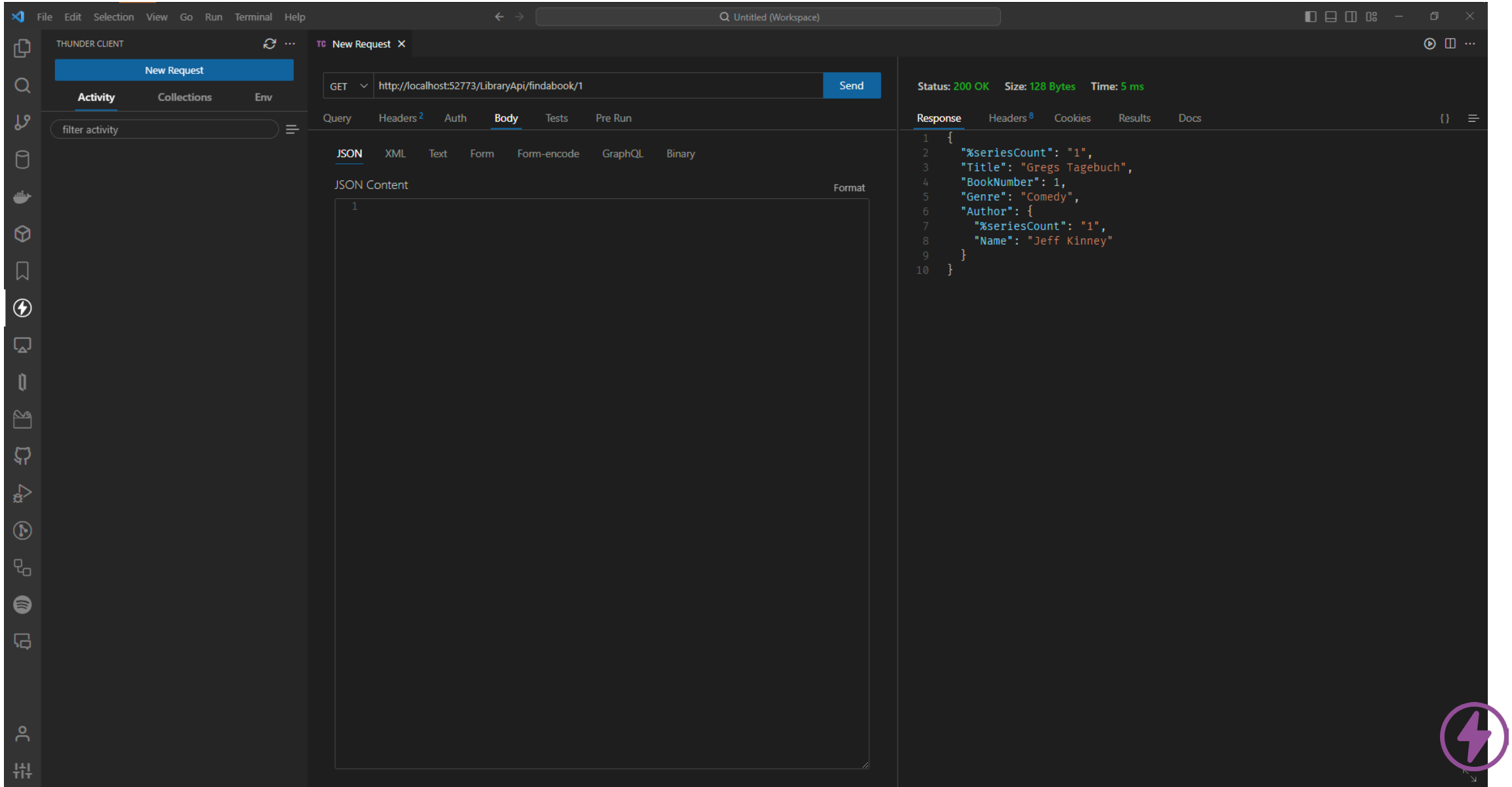
InterSystems®
DACH SYMPOSIUM 2023

Verifizieren

Tool: Thunder Client

Tool: SQLTools

Test /findabook



```
    name: Apache 2.0
    url: http://www.apache.org/licenses/LICENSE-2.0.html
schemes:
  - https
paths:
  /findabook/{bookid}:
    get:
      description: Find a book
      operationId: FindingBook
      produces:
        - application/json
      parameters:
        - name: bookid
          in: path
          type: string
          required: true
      responses:
        '200':
          description: Success
        '500':
          description: Server error
  /addfriend:
    post:
      description: Add a new friend
      operationId: AddingFriend
      produces:
        - application/json
      responses:
        '200':
          description: Success
```

An API for Lib

Apache 2.0

Schemes

HTTPS

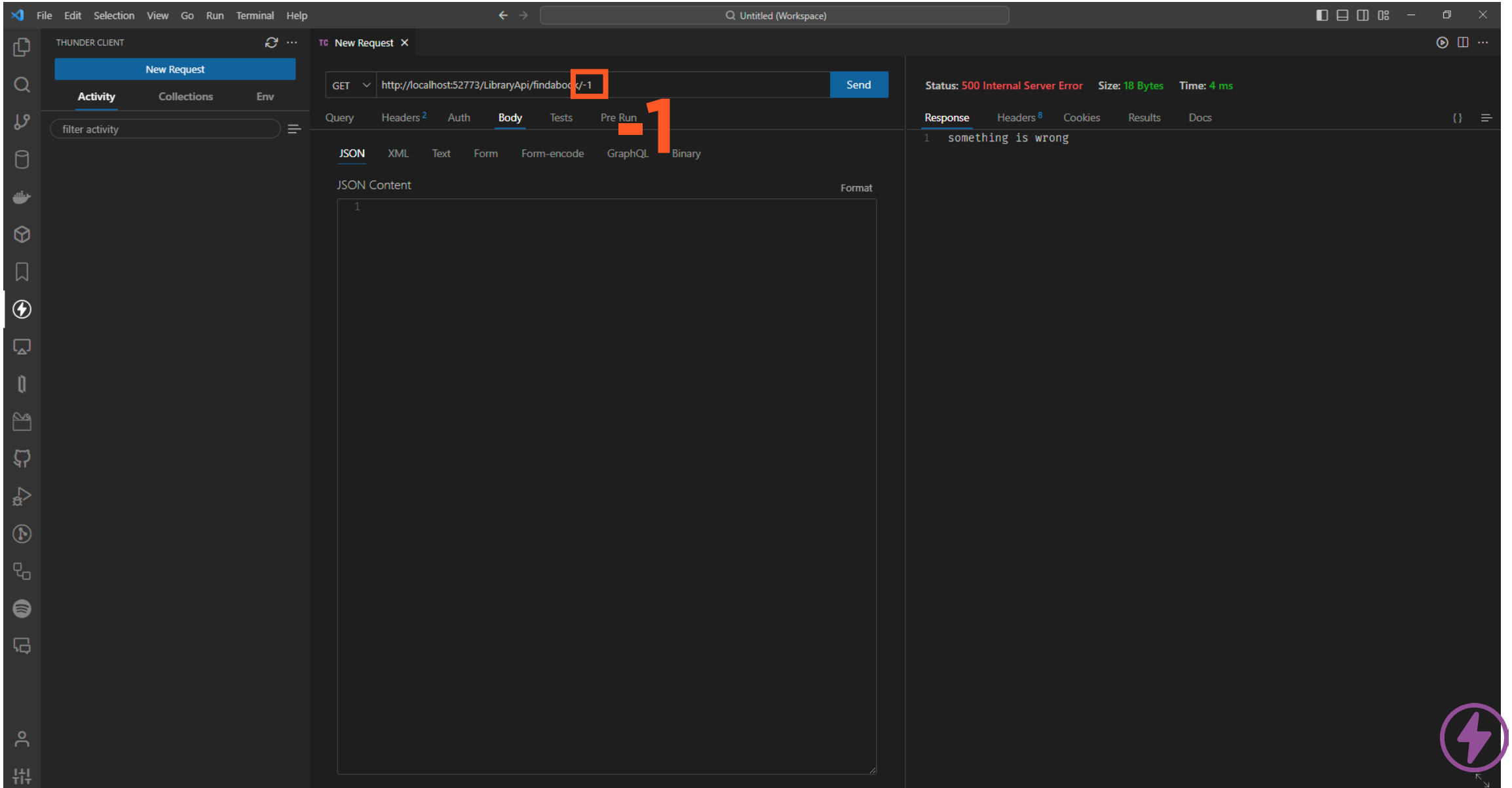
defaul

GET

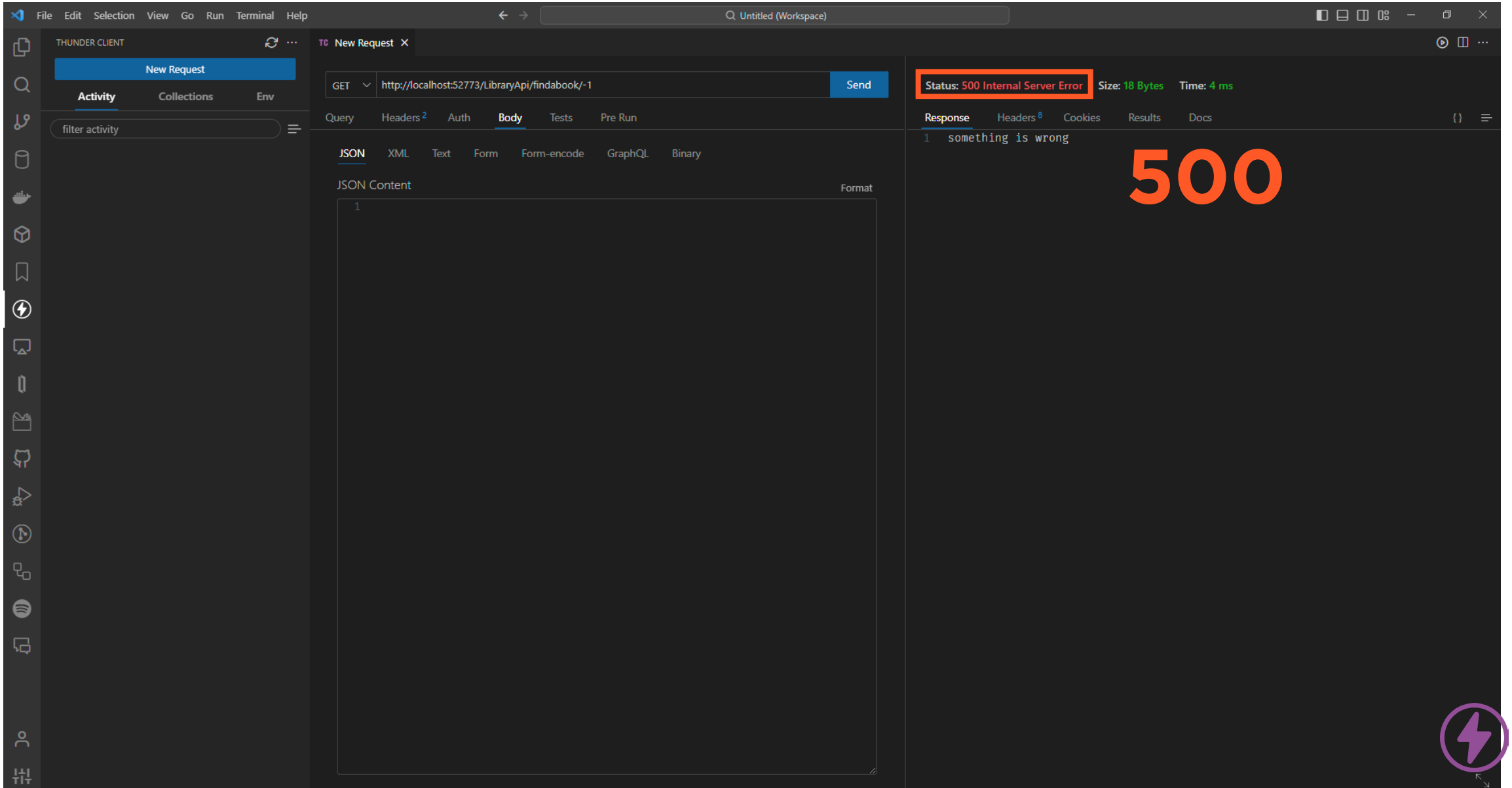
POST

Add a new

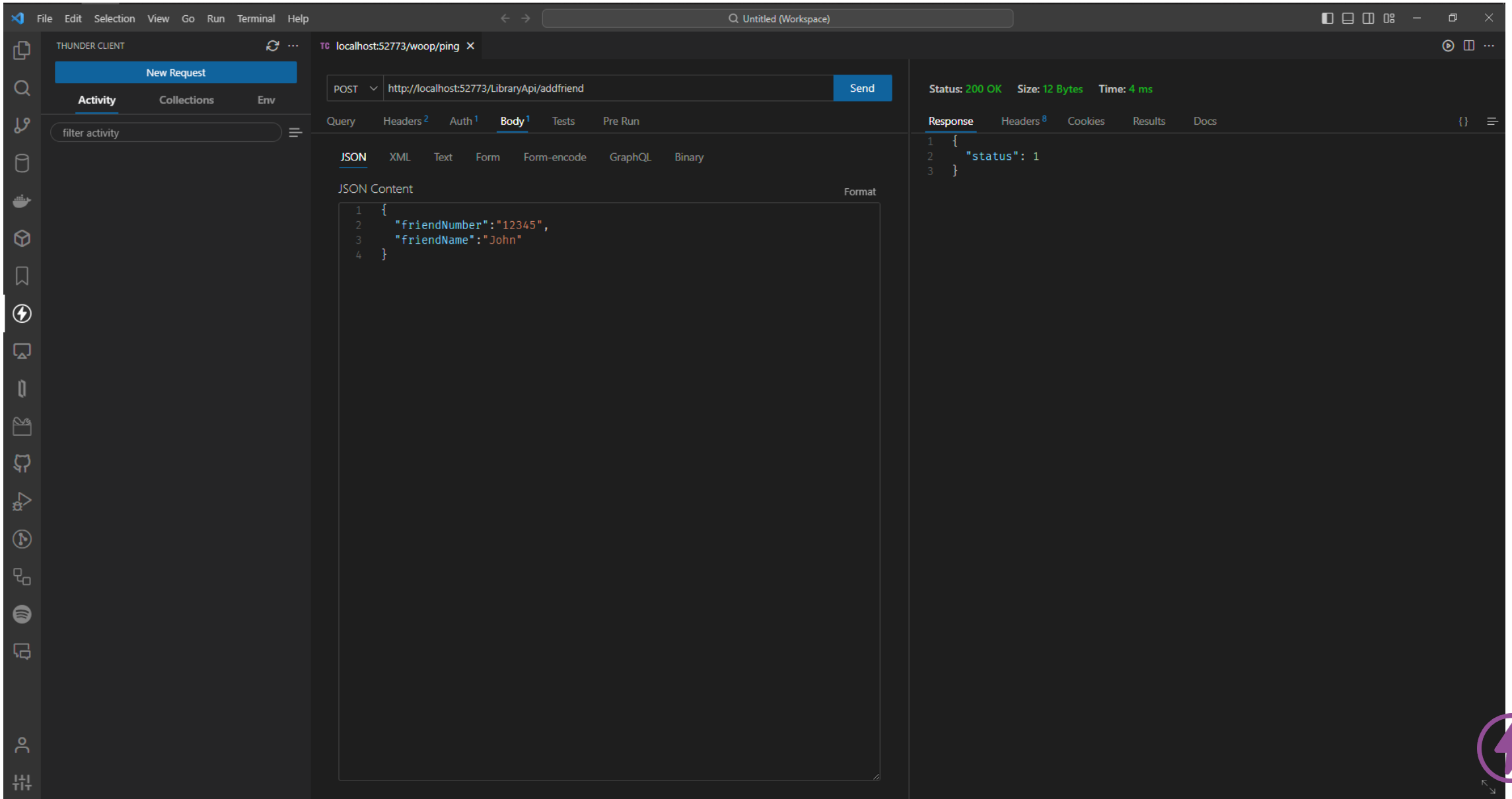
Test /findabook ungültige bookID



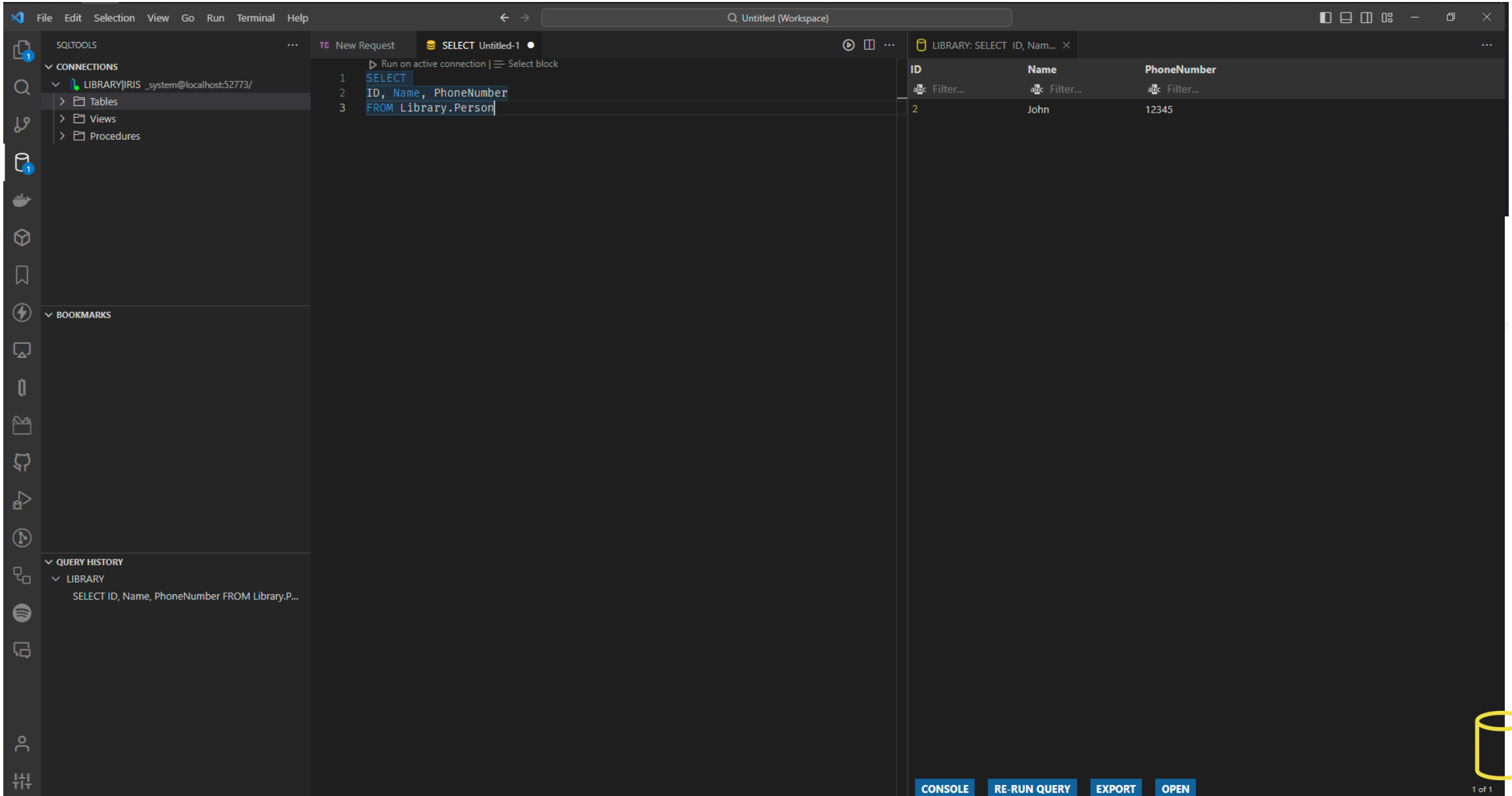
Test /findabook ungültige bookID



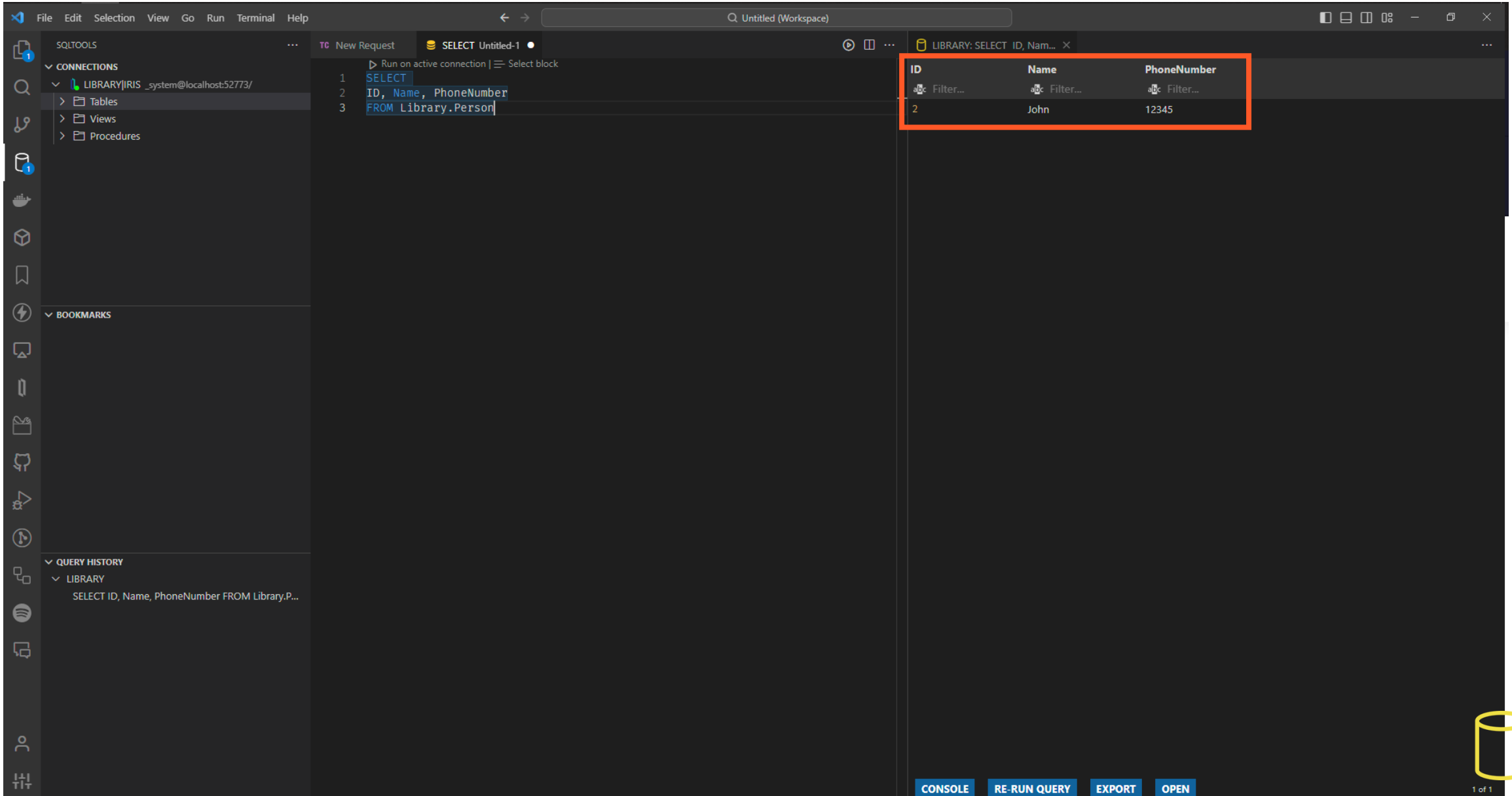
Test /addfriend



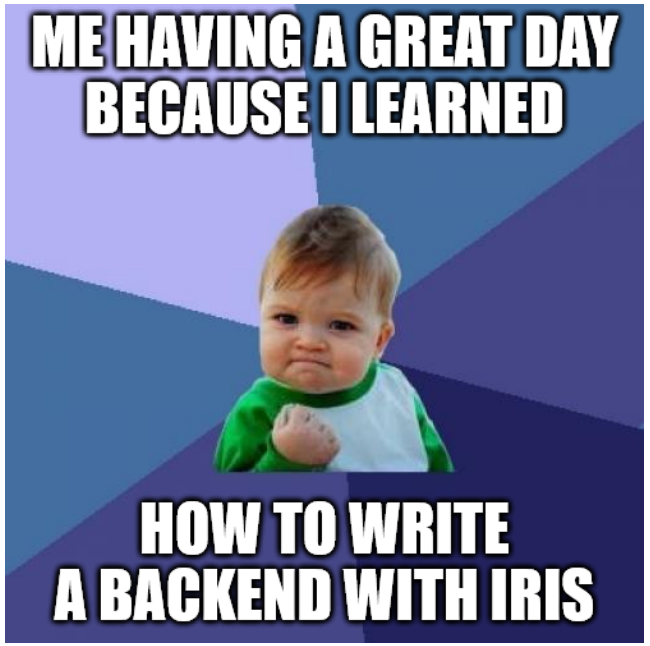
Test /addfriend SQL



Test /addfriend SQL



Thank you!


ME HAVING A GREAT DAY
BECAUSE I LEARNED
HOW TO WRITE
A BACKEND WITH IRIS

Tools