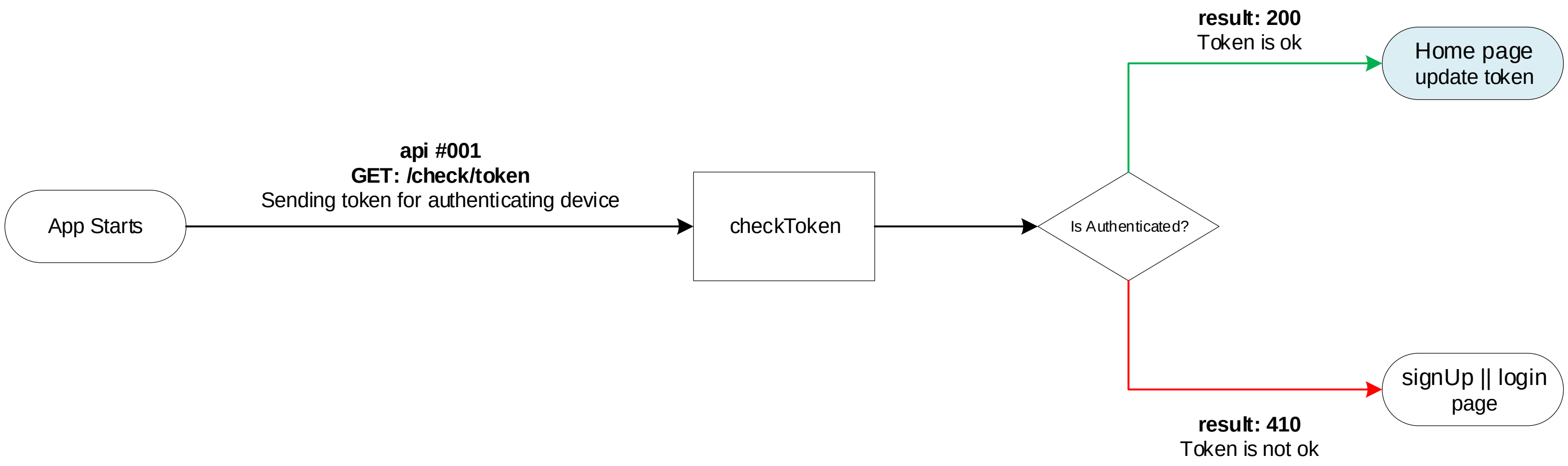
Application: Checking Token on start



App Starts

api #001
GET: /check/token
Sending token for authenticating device

checkToken

Is Authenticated?

result: 200
Token is ok

Home page
update token

result: 410
Token is not ok

signUp || login
page

api #001 : sapi/check/token

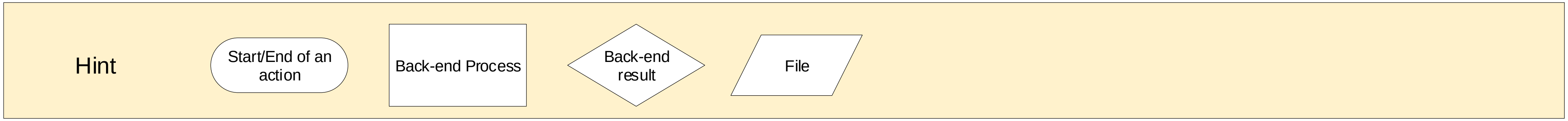
modulePath:
application/router/routes/check/get_check_token

type: GET

heders: {
    token: saved token on device, user’s token
}

body: {
}

results:
    200: token is ok + (JSON) [{ token: token }]
    410: token is not ok



Hint

Start/End of an
action

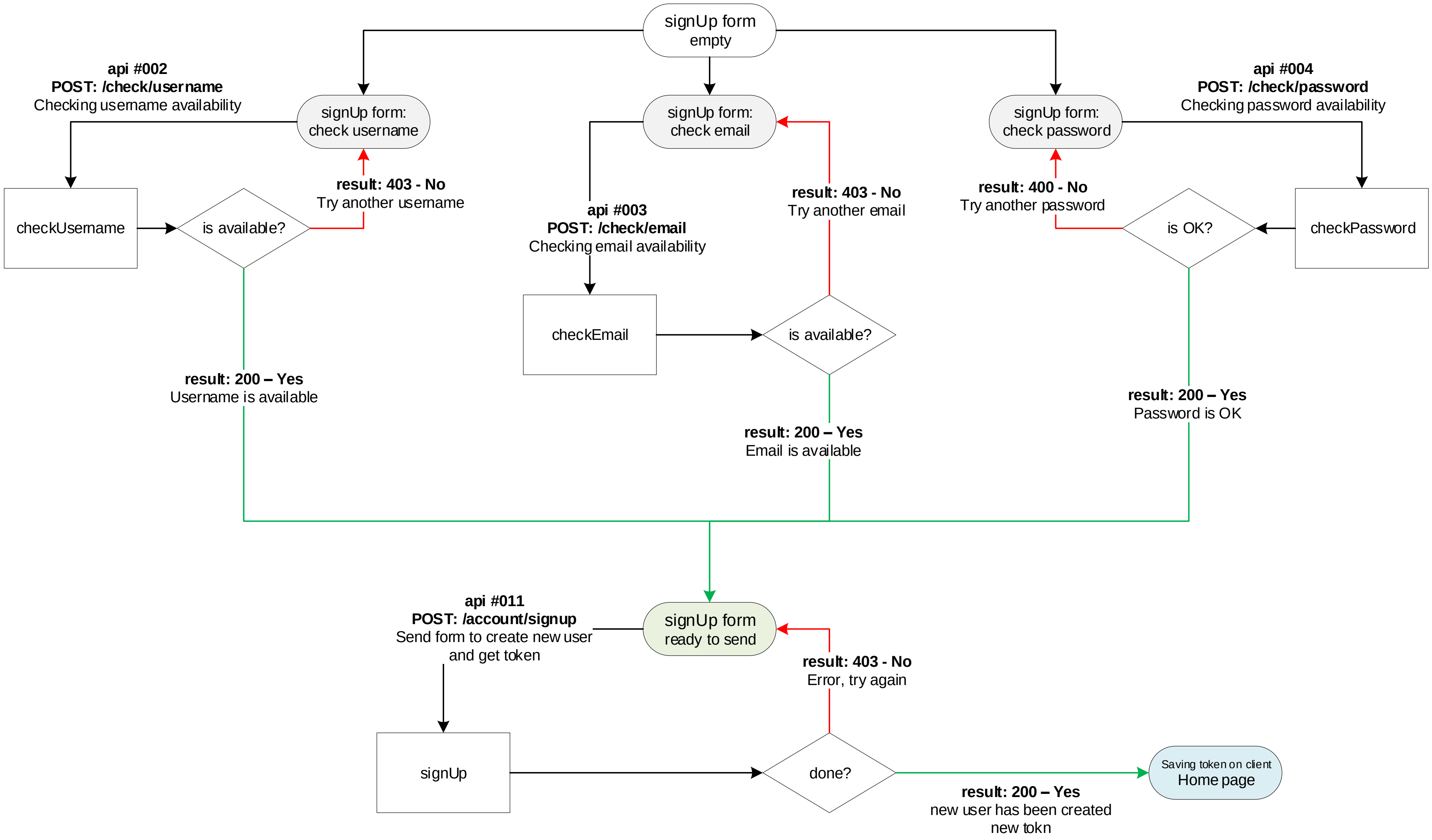
Back-end Process

Back-end
result

File

page 2-1

Sign Up



signUp form empty

api #002
POST: /check/username
Checking username availability

signUp form: check username

checkUsername

is available?

result: 403 - No
Try another username

result: 200 – Yes
Username is available

signUp form: check email

api #003
POST: /check/email
Checking email availability

checkEmail

is available?

result: 403 - No
Try another email

result: 200 – Yes
Email is available

signUp form: check password

api #004
POST: /check/password
Checking password availability

checkPassword

is OK?

result: 400 - No
Try another password

result: 200 – Yes
Password is OK

signUp form ready to send

api #011
POST: /account/signup
Send form to create new user
and get token

signUp

done?

result: 403 - No
Error, try again

result: 200 – Yes
new user has been created
new tokn

Saving token on client
Home page

page 2-2

Sign Up

api #002 :   api/check/username

modulePath:
application/router/routes/check/post_check_username

type: POST

heders: {
}

body: {
    username: inserted by user
}

results:
    200: new username is available
    403: username is not available

api #003 :   api/check/email

modulePath:
application/router/routes/check/post_check_email

type: POST

heders: {
}

body: {
    email: inserted by user
}

results:
    200: email is available
    403: email is not available

api #004 :   api/check/password

modulePath:
application/router/routes/check/post_check_email

type: POST

heders: {
}

body: {
    password: inserted by user
}

results:
    200: password is ok
    400: password is not ok

api #011 :   api/account/signup

modulePath:
application/router/routes/account/post_account_signup

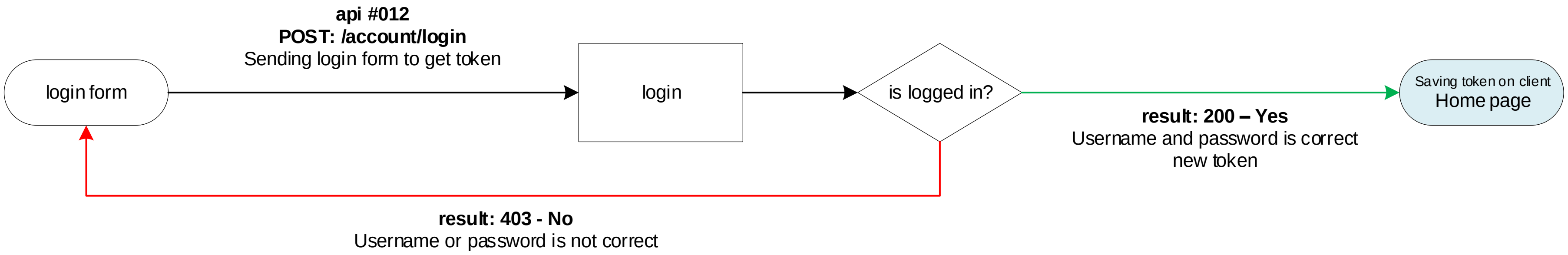
type: POST

heders: {
}

body: {
    username: inserted by user,
    password: inserted by user,
    email: inserted by user
}

results:
    200: new user has been created + (JSON) [{ token: token }]
    403: username is not available

Login



login form

api #012
POST: /account/login
Sending login form to get token

login

is logged in?

result: 200 – Yes
Username and password is correct
new token

Saving token on client
Home page

result: 403 - No
Username or password is not correct

api #012 : api/account/login

modulePath:
application/router/routes/account/post_account_login

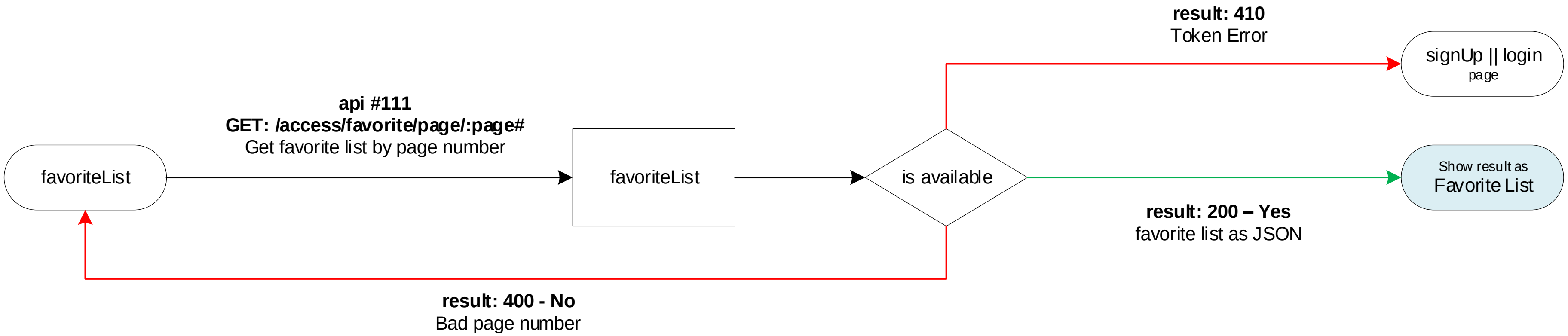
type: POST

heders: {
}

body: {
    username: inserted by user,
    password: inserted by user
}

results:
    200: username and password is correct + (JSON) [{ token: token }]
    403: username and password is not correct

User access: Favorite List



api #111 :   sapi/access/favorites

modulePath:
application/router/routes/access/post_access_favorites

type: post

heders: {
    token: user’s token
}

body: {
    pageNumber: Number
}

results:
    200: favorite list as JSON
    400: Bad page number
    410: Token is not available

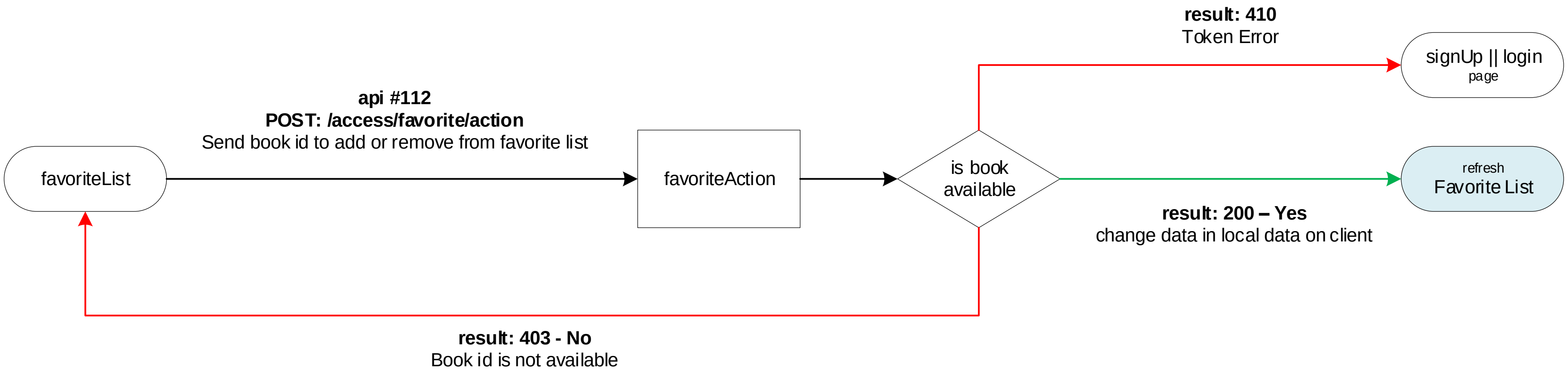
*** **/:page#** is the number of page of results, starts from 0
    each page contains up to 10 items if available

*** **api example:**   /access/favorite/page/2

*** **result example:**

```
[ {
pageCount: 54,
pageNumber: 2,
favoriteList:[{“_id”: “vsf8v76df8v4df”, “name”: “aaa”, “pageCount”: “120”, “uploadedBy”: “objectId”, “showOnSearch”: “true”, “uploadDate”: “Date”}, …]
} ]
```

User access: add to / remove from Favorite List



result: 410
Token Error

signUp || login
page

api #112
POST: /access/favorite/action
Send book id to add or remove from favorite list

favoriteList

favoriteAction

is book
available

refresh
Favorite List

result: 200 – Yes
change data in local data on client

result: 403 - No
Book id is not available

api #112 : sapi/access/favorite

modulePath:
application/router/routes/access/post_access_favorite

type: POST

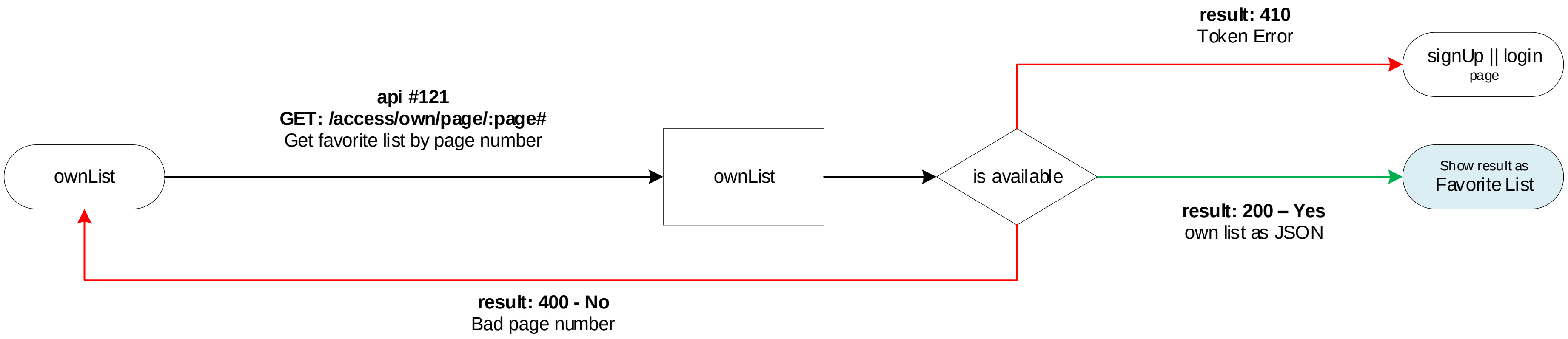
heders: {
    token: user’s token
}

body: {
    id: selected book _id from favorite list or search result
}

results:
    200: book has been added to / removed from favorite list
    403: Book id is not available
    410: Token is not available

*** **result example:**
        [{“_id”: “vsf8v76df8v4df”, “favorite”: “true”}]

User access: my books



api #121 : sapi/access/mybooks

modulePath:
application/router/routes/access/post_access_mybooks

type: GET

heders: {
    token: user's token
}

body: {
    pageNumber: Number
}

results:
    200: own list as JSON
    400: Bad page number
    410: Token is not available

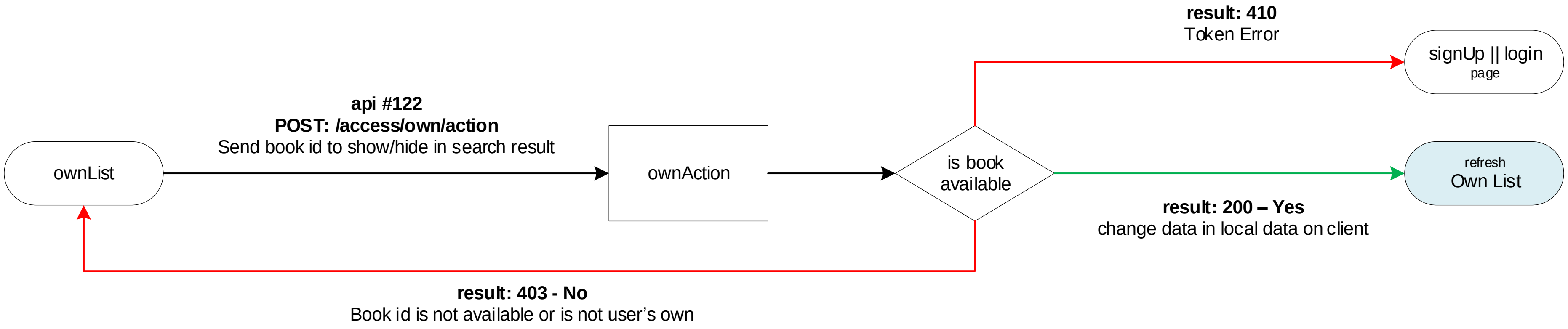
*** **/:page#** is the number of page of results, starts from 0
    each page contains up to 10 items if available

*** **api example:** /access/own/page/2

*** **result example:**

```
[ {
pageCount: 54,
pageNumber: 2,
myBookList:[{"_id": "vsf8v76df8v4df", "name": "aaa", "pageCount": "120", "uploadedBy": "objectId", "showOnSearch": "true", "uploadDate": "Date"}, …]
} ]
```

User access: hide from / show in Search Result


ownList
api #122
POST: /access/own/action
Send book id to show/hide in search result
ownAction
is book available
result: 410
Token Error
signUp || login page
refresh Own List
result: 200 – Yes
change data in local data on client
result: 403 - No
Book id is not available or is not user's own

api #122 : sapi/access/showable

modulePath:
application/router/routes/access/post_access_showable

type: POST

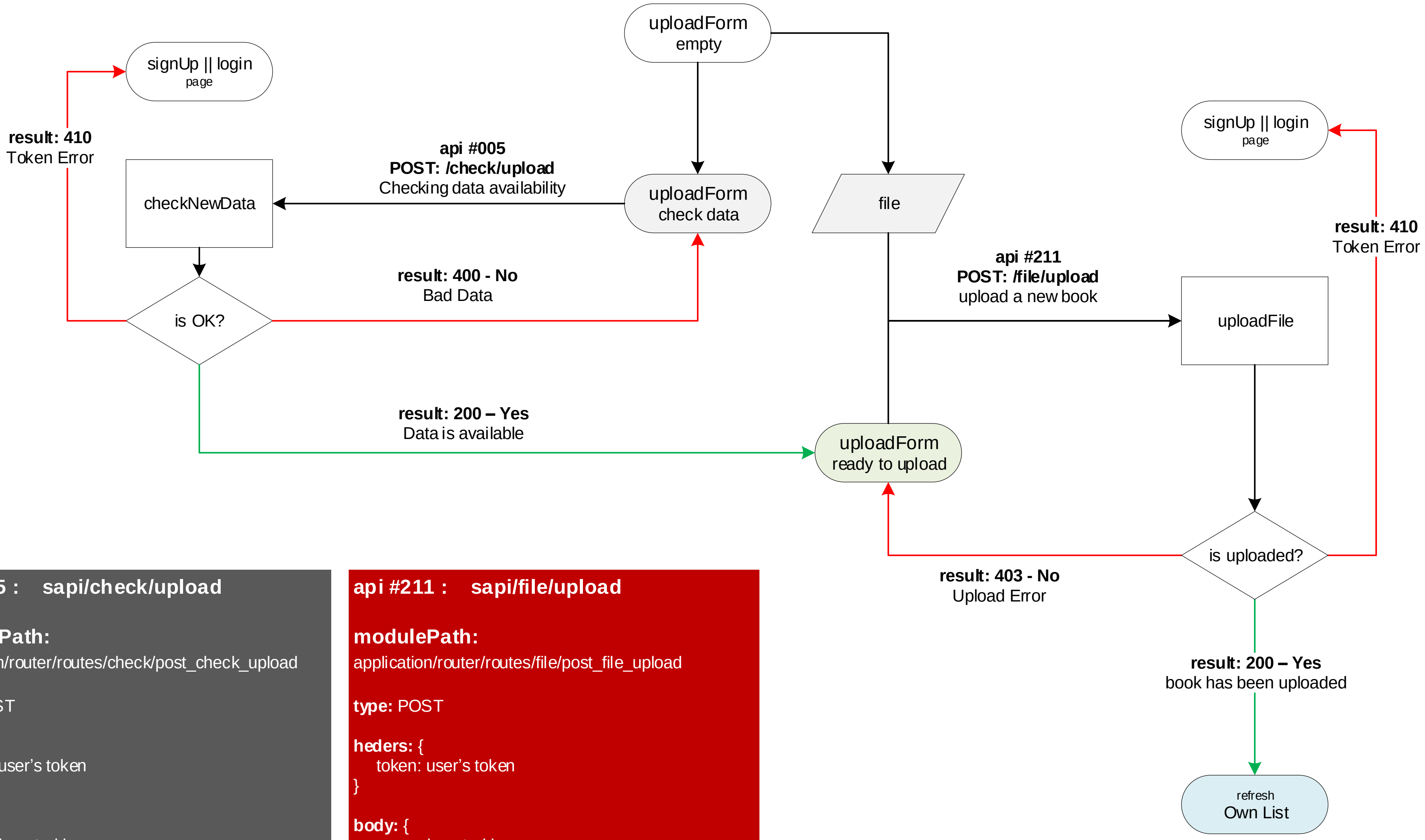
heders: {
token: user's token
}

body: {
id: selected book _id from own list or search result
}

results:
200: book has been added to / removed from favorite list
403: Book id is not available or is not user's own
410: Token is not available

*** **result example:**
[{"_id": "vsf8v76df8v4df", "showOnSearch": "false"}]

File: upload a new book (not exist)



uploadForm
empty

uploadForm
check data

api #005
POST: /check/upload
Checking data availability

checkNewData

is OK?

result: 410
Token Error

signUp || login
page

result: 400 - No
Bad Data

result: 200 – Yes
Data is available

file

api #211
POST: /file/upload
upload a new book

uploadFile

uploadForm
ready to upload

is uploaded?

result: 410
Token Error

signUp || login
page

result: 403 - No
Upload Error

result: 200 – Yes
book has been uploaded

refresh
Own List

api #005 : sapi/check/upload

modulePath:
application/router/routes/check/post_check_upload

type: POST

heders: {
token: user’s token
}

body: {
name: inserted by user,
pageCount: insert by user
}

results:
200: data is ok
400: data is not ok
410: Token is not available

api #211 : sapi/file/upload

modulePath:
application/router/routes/file/post_file_upload

type: POST

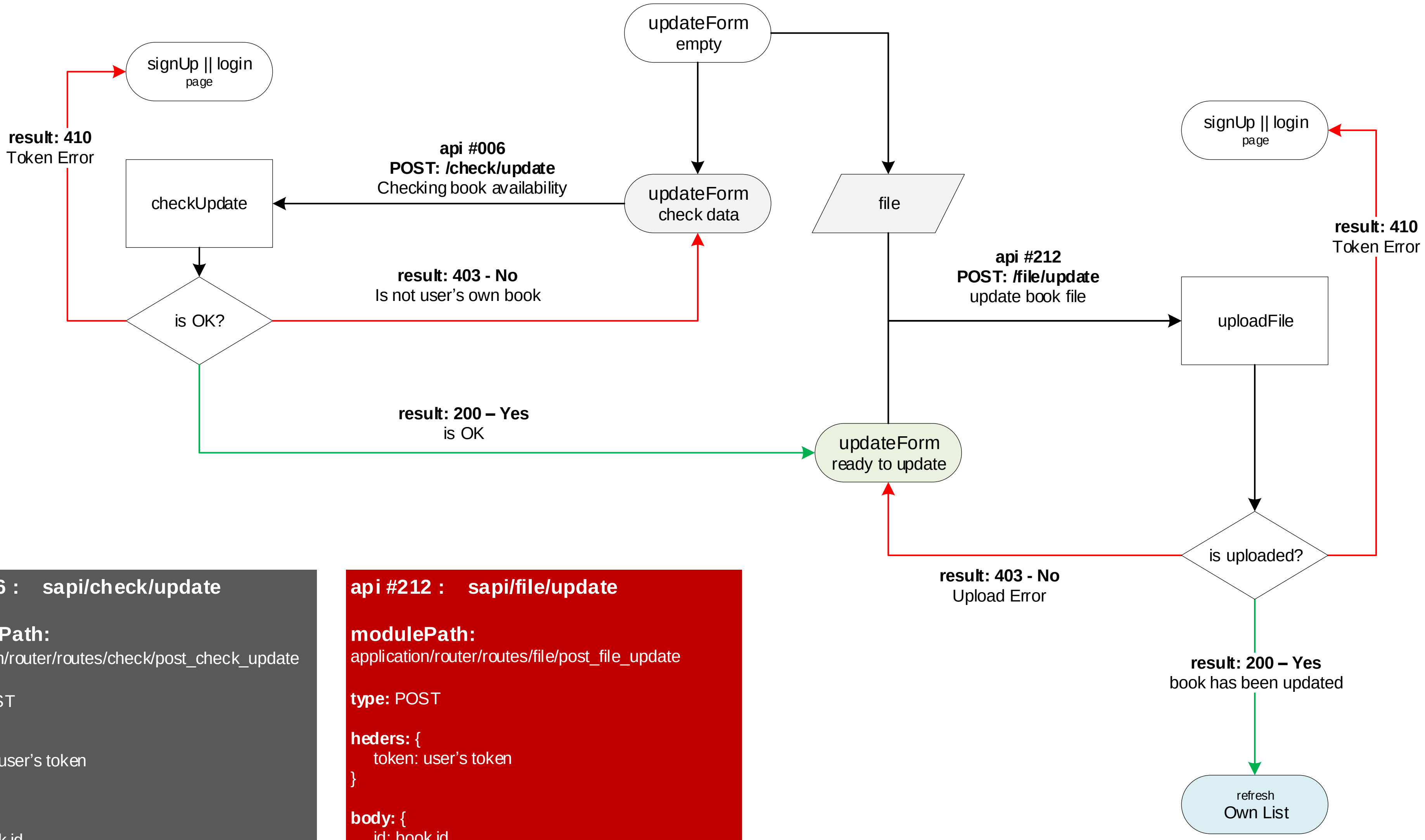
heders: {
token: user’s token
}

body: {
name: inserted by user,
pageCount: inserted by user
}

results:
200: Book has been uploaded
400: No file or Data
403: Upload Error, bad file type or size
410: Token is not available

page 9

File: update the file of a book (not exist)



updateForm
empty

updateForm
check data

api #006
POST: /check/update
Checking book availability

checkUpdate

is OK?

result: 410
Token Error

signUp || login
page

result: 403 - No
Is not user's own book

result: 200 – Yes
is OK

updateForm
ready to update

file

api #212
POST: /file/update
update book file

uploadFile

is uploaded?

result: 410
Token Error

signUp || login
page

result: 403 - No
Upload Error

result: 200 – Yes
book has been updated

refresh
Own List

api #006 : sapi/check/update

modulePath:
application/router/routes/check/post_check_update

type: POST

heders: {
    token: user's token
}

body: {
    id: book id
}

results:
    200: is own, available to update
    403: is not user's own
    410: Token is not available

api #212 : sapi/file/update

modulePath:
application/router/routes/file/post_file_update

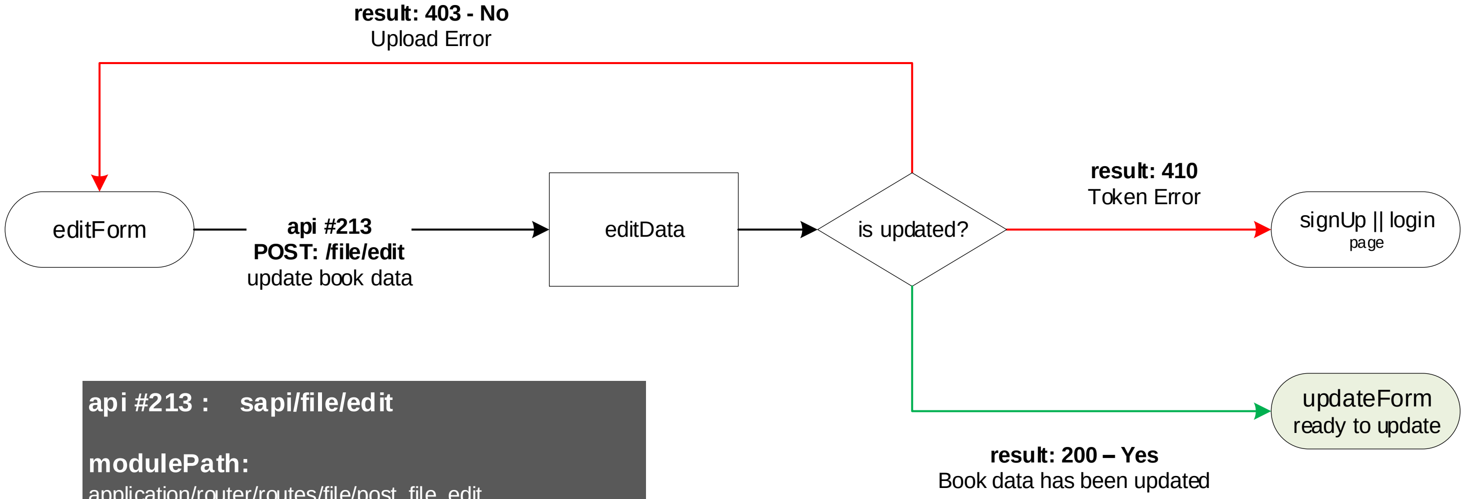
type: POST

heders: {
    token: user's token
}

body: {
    id: book id
}

results:
    200: Book has been updated
    400: No file or Data
    403: Upload Error, bad file type or size
    410: Token is not available

File: edit data of a book



result: 403 - No
Upload Error

editForm

api #213
POST: /file/edit
update book data

editData

is updated?

result: 410
Token Error

signUp || login
page

updateForm
ready to update

result: 200 – Yes
Book data has been updated

api #213 : sapi/file/edit

modulePath:
application/router/routes/file/post_file_edit

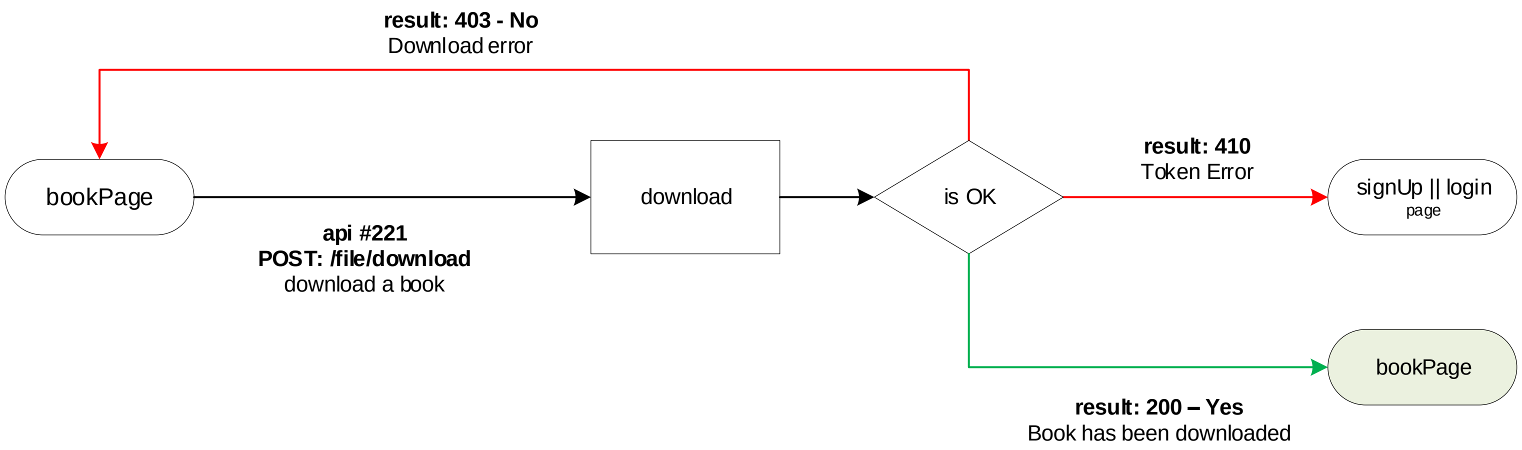
type: POST

heders: {
    token: user's token
}

body: {
    id: book id,
    name: new or old name inserted by user,
    pageCount: new or old pageCount inserted by user
}

results:
    200: Book data has been edited
    400: no change
    403: Not an own book or bad id
    410: Token is not available

File: download a book (not exist)



result: 403 - No
Download error

bookPage

api #221
POST: /file/download
download a book

download

is OK

result: 410
Token Error

signUp || login
page

bookPage

result: 200 – Yes
Book has been downloaded

api #221 :   sapi/file/download

modulePath:
application/router/routes/file/post_file_download

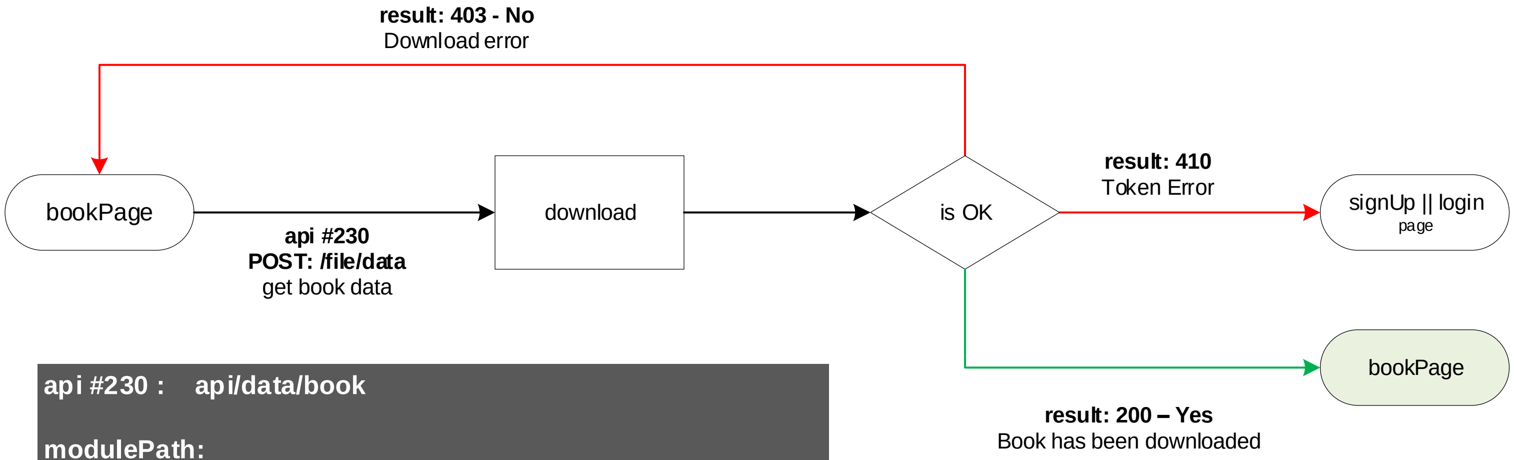
type: POST

heders: {
    token: user’s token
}

body: {
    id: book id
}

results:
    200: Book has been downloaded
    403: Bad id, book is not available
    410: Token is not available

File: Book Data View



api #230 : api/data/book

modulePath:
application/router/routes/data/post_data_book

type: POST

heders: {
}

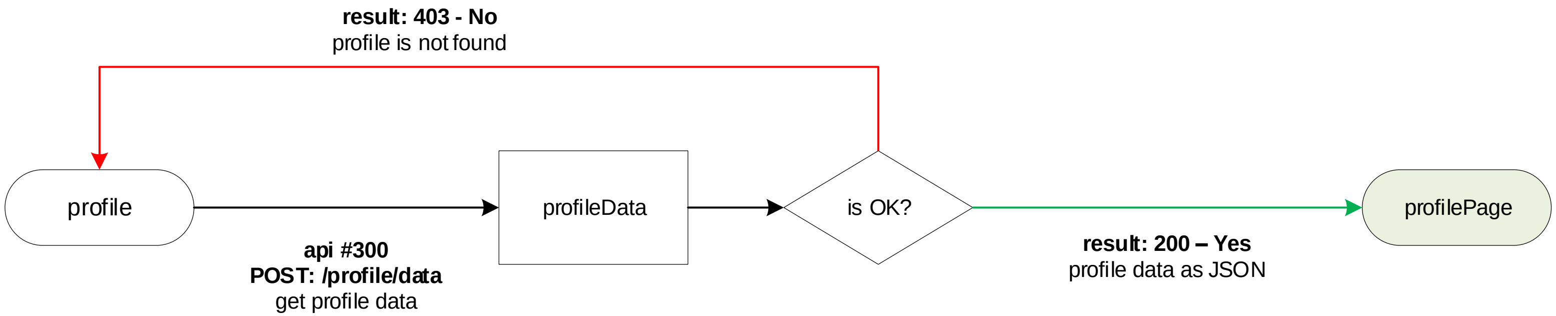
body: {
    id: book id
}

results:
    200: Data as JSON
    403: Bad id, book is not available

*** **result example:**

```
               [{“_id”: “vsf8v76df8v4df”,
                “name”: “aaa”,
                “pageCount”: “120”,
                “uploadedBy”: “objectId”,
                “uploaderUsername”: “username”,
                “ownFavorite”: “false”,
                “uploadDate”: “Date”,
                “updateDate”: “Date”,
                “favoriteCount”: “Number”, //how many users have this in favorite list
                “downloadCount”: “Number”
```

Profile: View Data



result: 403 - No
profile is not found

profile

api #300
POST: /profile/data
get profile data

profileData

is OK?

result: 200 – Yes
profile data as JSON

profilePage

api #300 :   /data/profile

modulePath:
application/router/routes/data/post_data_profile

type: POST

heders: {
    token: user’s token
}

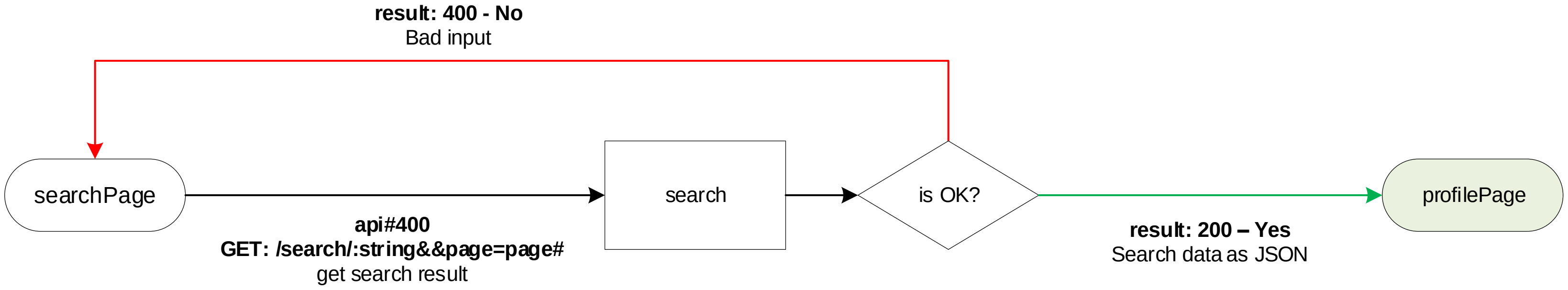
body: {
    id: user id
}

results:
    200: Data as JSON
    403: Bad id, user is not available

*** **result example:**

```
[{“_id”: “vsf8v76df8v4df”,
 “username”: “farbod”,
 “createDate”: “Date”,
 “lastSigned”: “Date”,
 “uploadedCount”: “Number”, //number of uploaded books
 “favoriteCount”: “Number”, //number of favorite books
 “ownProfile”: “false”, //true if token is matched to profile
}, …]
```

Search



result: 400 - No
Bad input

searchPage

search

is OK?

profilePage

api#400
GET: /search/:string&&page=page#
get search result

result: 200 – Yes
Search data as JSON

api #400 : api/data/search

modulePath:
application/router/routes/data/post_data_search

type: GET

heders: {
}

body: {
search: String,
pageNumber: search page number,
sortOrde: date/pageCount
}

results:
200: Data as JSON
403: Bad input

*** **page=page#** is the number of page of results, starts from 0
each page contains up to 10 items if available

***** request example:**
/search/bookname&&page=0 it means searching for title “bookname” in first page of list

*** **result example:**
[{
searchPageCount: 54,
searchPage: 2,
searchResult: [{“_id”: “vsf8v76df8v4df”, “name”: “aaa”, “pageCount”: “120”, “uploadedBy”: “objectId”, “showOnSearch”: “true”, “uploadDate”: “Date”}, …]
}]