

# Създаване на приложения на една страница (SPA) с jQuery AJAX, REST и Kinvey



# The Book Library Project

A screenshot of a web browser displaying the 'books.html' page of the Book Library Project. The page has a header with navigation links and a welcome message. Below the header is a table listing books with columns for Title, Author, Description, and Actions. The footer contains a copyright notice.

← → C Q books.html

Home List Books Create Book Logout Welcome, guest!

**Books**

Title	Author	Description	Actions
Java Fundamentals	Bay Ivan	Strong Java book. Not for beginners	
Software Technologies	Some author  	Creating simple apps with HTML5, PHP, C# and Java.	
C# Basics	Some author  	Very good C# book!	
Some title  	Some author  	Some description  	<a href="#">[Delete]</a> <a href="#">[Edit]</a>

Book Library - Simple SPA Application

# The "Book Library" App

- ▶ Проектирайте и създайте приложение за библиотека от една страница (SPA) в HTML5 с REST back-end в Kinvey
  - ▶ Книгите трябва да имат заглавие, автор и описание (title, author and description)
- ▶ Внедрете следните функционалности :
  - ▶ Вход, регистрация, изход, всички книги, създаване нова книга, редактирайте съществуваща, изтриване
  - ▶ Всеки може да разгледа всички книги
  - ▶ Само създателят на книгата може да редактира / изтрива собствените си книги

# Началния екран



Welcome 

Welcome to our book library.

Book Library - Simple SPA Application

# Екран за вход

The screenshot shows a web browser window with the URL "books.html" in the address bar. The main content is a login form for a "Book Library - Simple SPA Application". The "Login" button is highlighted with a green border. A large green arrow points from the text "Please login" to the "Login" button. The "Username" field contains "maria" and the "Password" field contains a series of dots (".....").

← → C books.html

Home Login Register

Please login

Username:

maria

Password:

.....

Login

Book Library - Simple SPA Application

# Екран за вход: Невалидно влизане



books.html

Home Login Register

Error: Invalid credentials. Please retry your request with  
correct credentials

Please login

Username:

invalid user

Password:

...

Login



Book Library - Simple SPA Application

# Еcran за регистрация

← → C books.html

Home Login Register

Please register here

Username:  
new\_user

Password:  
.....

Register

Book Library - Simple SPA Application

# Екран на списъка с книги

A screenshot of a web browser window displaying a simple Single Page Application (SPA) for managing books. The URL bar shows 'books.html'. The header includes links for 'Home', 'List Books' (which is highlighted with a green border), 'Create Book', and 'Logout', along with a welcome message 'Welcome, guest!'. A green success message 'Books loaded.' is displayed above the main content area. The main content area is titled 'Books' and contains a table listing five books with columns for Title, Author, Description, and Actions.

Title	Author	Description	Actions
Java Fundamentals	Bay Ivan	Strong Java book. Not for beginners	
Software Technologies	Bay Ivan	Creating simple apps with HTML5, PHP, C# and Java.	
C# Basics	Bay Ivan	Very good C# book!	
Some title  	Bay Ivan	Some text	<a href="#">[Delete]</a> <a href="#">[Edit]</a>
Some title  	Some author  	Some description  	<a href="#">[Delete]</a> <a href="#">[Edit]</a>

# Екран за създаване на книга

← → ⌛

Home List Books **Create Book** Logout Welcome, guest!

**Create new book** 

Title:

Author:

Description:  
New book description  
comes here...

Book Library - Simple SPA Application

# Екран за редактиране на книга

books.html

Home List Books Create Book Logout  
Welcome, guest!

## Books

Title	Author	Description	Actions
Some title  	Some author  	Very good C# book!	[Delete]
Some title  	Some author  	Some description  	[Delete] [Edit]

Book Library - Simple SPA Application

books.html

Home List Books Create Book Logout Welcome, guest!

### Edit existing book

Title:  
Some title <br>

Author:  
Some author <br>

Description:  
Some description <br>

Edit



Book Library - Simple SPA Application

# Еcran за изтриване на книга

books.html

Home List Books Create Book Logout Welcome, guest!

## Books

Title	Author	Description	
Java Fundamentals	Bay Ivan	Strong Java book. Not for beginners	
xxx	xxx	Creating simple apps with HTML5, PHP, C# and Java.	
xxx	xxx	Very good C# book!	
xxx	xxx	Some text	<a href="#">[Delete]</a> <a href="#">[Edit]</a>
Some title  	Some author  	Some description  	<a href="#">[Delete]</a> <a href="#">[Edit]</a>
xxx	xxx	xxx	<a href="#">[Delete]</a> <a href="#">[Edit]</a>

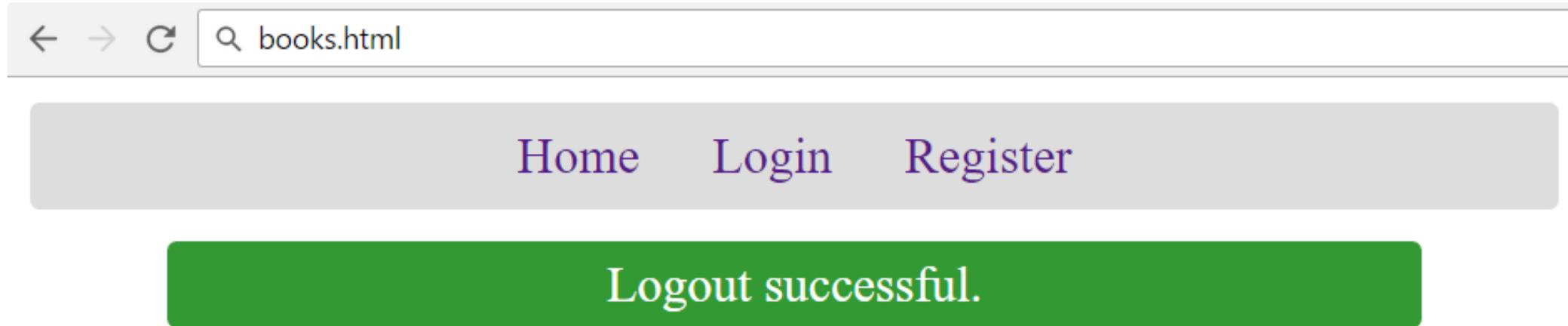
books.html

Home List Books Create Book Logout Welcome, guest!

Books deleted.



# Екран за излизане

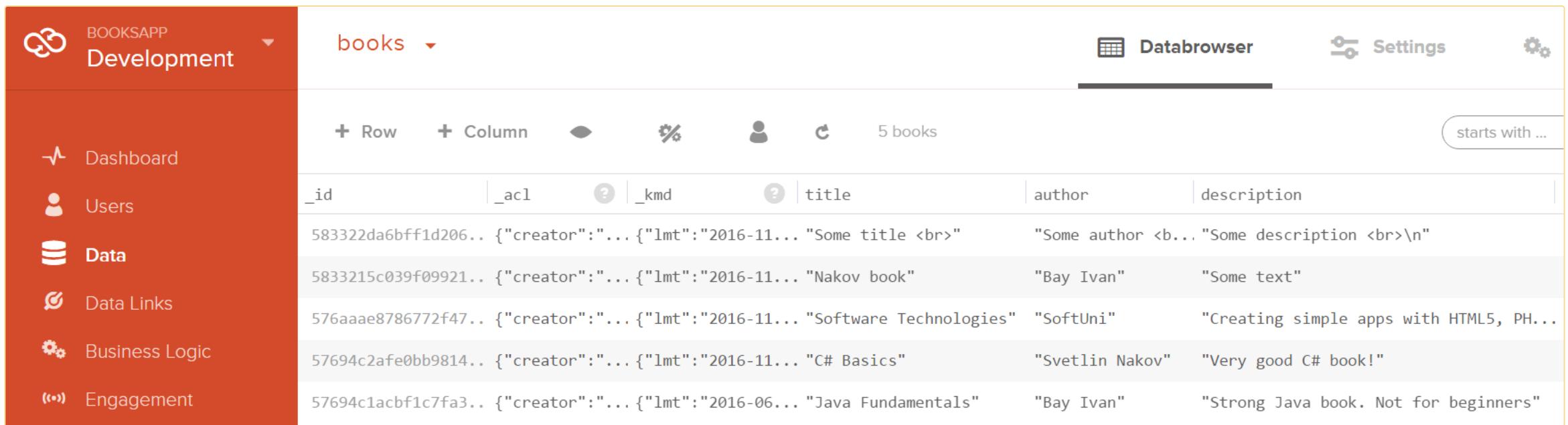


## Welcome

Welcome to our book library.

Book Library - Simple SPA Application

# The Kinvey-Based Back-End



The screenshot shows the Kinvey Databrowser interface for a 'books' collection. The left sidebar has a red background with icons and labels: Dashboard, Users, Data (selected), Data Links, Business Logic, and Engagement. The main area has a white background with a header bar containing the collection name 'books', a 'Databrowser' button, a 'Settings' button, and a search input field. Below the header is a toolbar with buttons for '+ Row', '+ Column', a refresh icon, a percentage icon, a user icon, and a circular arrow icon. The text '5 books' is displayed. A table below shows five rows of book data with columns: \_id, \_acl, \_kmd, title, author, and description. The data includes titles like 'Some title', 'Nakov book', 'Software Technologies', 'C# Basics', and 'Java Fundamentals', along with authors like 'Some author', 'Bay Ivan', 'SoftUni', 'Svetlin Nakov', and 'Bay Ivan', and descriptions.

_id	_acl	_kmd	title	author	description
583322da6bff1d206...	{"creator": "...", "lmt": "2016-11-15T12:00:00.000Z", "modified": "2016-11-15T12:00:00.000Z", "version": 1}		"Some title  "	"Some author <b>...</b>"	"Some description  \n"
5833215c039f09921...	{"creator": "...", "lmt": "2016-11-15T12:00:00.000Z", "modified": "2016-11-15T12:00:00.000Z", "version": 1}		"Nakov book"	"Bay Ivan"	"Some text"
576aaae8786772f47...	{"creator": "...", "lmt": "2016-11-15T12:00:00.000Z", "modified": "2016-11-15T12:00:00.000Z", "version": 1}		"Software Technologies"	"SoftUni"	"Creating simple apps with HTML5, PH..."
57694c2afe0bb9814...	{"creator": "...", "lmt": "2016-11-15T12:00:00.000Z", "modified": "2016-11-15T12:00:00.000Z", "version": 1}		"C# Basics"	"Svetlin Nakov"	"Very good C# book!"
57694c1acbf1c7fa3...	{"creator": "...", "lmt": "2016-06-15T12:00:00.000Z", "modified": "2016-06-15T12:00:00.000Z", "version": 1}		"Java Fundamentals"	"Bay Ivan"	"Strong Java book. Not for beginners"

Users and Books Collections

# Създай Kinvey App

- ▶ Как да създадеш профил, може да видиш [тук](#).

The screenshot shows a web browser window with the URL <https://console.kinvey.com/apps/new>. The page title is "Create your new app backend". The navigation bar includes links for "My Apps", "New App" (which is underlined), and "Profile". A large input field contains the text "BooksApp". Below it is a prominent green button labeled "Create App".

← → ⌂ <https://console.kinvey.com/apps/new>

My Apps    **New App**    Profile

---

## Create your new app backend

Great, let's get your backend up and running! Just one question:

What do you want to call this app?

**Create App**

# Създай колекция Books

New Collection

books ▾

+ Row + Column ⚡ % ⚡ 5 books starts with ...

_id	_acl	_kmd	title	author	description
583322da6bff1d206...	{"creator": "...", "lmt": "...", "Some title  "}		"Some author  "	"Some description  \n"	
5833215c039f09921...	{"creator": "...", "lmt": "...", "Nakov book"}		"Bay Ivan"	"Some text"	
576aaae8786772f47...	{"creator": "...", "lmt": "...", "Software Technologies"}		"SoftUni"	"Creating simple apps with HTML5, PHP, C# and Java." ...ering,	
57694c2afe0bb9814...	{"creator": "...", "lmt": "...", "C# Basics"}		"Svetlin Nakov"	"Very good C# book!"	
57694c1acbf1c7fa3...	{"creator": "...", "lmt": "...", "Java Fundamentals"}		"Bay Ivan"	"Strong Java book. Not for beginners"	

▶ Колони: title + author + description

No other data sources available. [Link to one now.](#)

[Create collection](#)

# Тествай Kinvey Back-End: Създаване на потребител

**POST**

[\*\*https://baas.kinvey.com/user/{app\\_id}/\*\*](https://baas.kinvey.com/user/{app_id}/)

**Authorization: Basic base64(app\_id:app\_secret)**

{"username": "todor", "password": "pass123"}

The screenshot shows the Postman interface with a successful API call to Kinvey's user registration endpoint.

**Request Details:**

- Method: POST
- URL: https://baas.kinvey.com/user/kid\_rkcLxcUr/
- Headers: Authorization (Basic base64(app\_id:app\_secret))
- Body (raw JSON):

```
1 {
2   "username": "todor",
3   "password": "pass123"
4 }
```

**Response Details:**

- Status: 201 Created
- Time: 364 ms
- Body (Pretty JSON):

```
1 {
2   "username": "todor",
3   "password": "pass123",
4   "_kmd": {
5     "lmt": "2016-11-22T09:09:05.080Z",
6     "ect": "2016-11-22T09:09:05.080Z",
7     "authToken": "5c3c23dc-c66a-4406-9dfb-ec7b3bdffc6d.6+xE23/eoid/eji9CgFiTqFe+LVCjBx19V7NqC50qh8="
8   }
}
```

# Вход на потребител

**POST**

[https://baas.kinvey.com/user/{app\\_id}/login](https://baas.kinvey.com/user/{app_id}/login)

**Authorization: Basic base64(app\_id:app\_secret)**

{"username": "todor", "password": "pass123"}

The screenshot shows a Postman interface with the following details:

- Method:** POST
- URL:** https://baas.kinvey.com/user/kid\_rkcLxcUr/login
- Headers:** Authorization (set to basic auth), Headers (2), Body (raw JSON), Pre-request Script, Tests (12).
- Body:** Raw JSON payload:

```
1 {
2   "username": "todor",
3   "password": "pass123"
4 }
```
- Response:** Status: 200 OK, Time: 229 ms. The response body is:

```
1 {
2   "_id": "58340b31f08321f70db381a4",
3   "username": "todor",
4   "kmd": {
5     "lmt": "2016-11-22T09:09:05.080Z",
6     "ect": "2016-11-22T09:09:05.080Z",
7     "authtoken": "8add6102-355c-4c61-a621-88c4fbbe7e8e./zHdsZWVSIIn0oxQrjnQNtUj2zjVbMv1rAh1YciSByik="
8 },
```

# Списък на всички книги

**GET** [https://baas.kinvey.com/appdata/{app\\_id}/books](https://baas.kinvey.com/appdata/{app_id}/books)

**Authorization:** Kinvey authtoken

The screenshot shows the Postman application interface. At the top, there is a search bar with the URL `https://baas.kinvey.com/appdata/kid_rkcLxcUr/books/`. Below the search bar are buttons for `Params`, `Send`, and `Save`. The `Headers (1)` tab is selected, showing a single entry for the `Authorization` header with the value `Kinvey dcf2f66a-48f8-4e50-8db2-fbfee6c1e2cb.+2MmXtaH1SdR7ey`. The `Body` tab is also visible at the bottom.

**Headers (1)**

key	value
Authorization	Kinvey dcf2f66a-48f8-4e50-8db2-fbfee6c1e2cb.+2MmXtaH1SdR7ey

**Body**

Status: 200 OK Time: 202 ms

Pretty Raw Preview JSON

```
1 [  
2 {  
3   "_id": "57694c1acbf1c7fa38033806",  
4   "title": "Java Fundamentals",  
5   "author": "Bay Ivan",  
6   "description": "Strong Java book. Not for beginners",
```

# Създай нова книга

**POST**

[https://baas.kinvey.com/appdata/{app\\_id}/books](https://baas.kinvey.com/appdata/{app_id}/books)

**Authorization:** Kinvey authtoken

{ "title": "ttt", "author": "aaa", "description": "ddd" }

The screenshot shows the Postman interface with the following details:

- Method:** POST
- URL:** https://baas.kinvey.com/appdata/kid\_rkcLxcUr/books/
- Headers:** (2) - Authorization (Kinvey authtoken)
- Body:** (raw, JSON application/json)  
Content:

```
1 { "title": "ttt", "author": "aaa", "description": "ddd" }
```
- Response:** Status: 201 Created, Time: 510 ms
- Body Content:**

```
1 [
2   {
3     "title": "ttt",
4     "author": "aaa",
5     "description": "ddd",
6     "_acl": {
7       "creator": "576911067f36dafb32bb60d1"
8     }
9   }
10 ]
```

# Test the Kinvey Back-End: Edit Book

PUT

[https://baas.kinvey.com/appdata/{app\\_id}/books/{id}](https://baas.kinvey.com/appdata/{app_id}/books/{id})

Authorization: Kinvey authtoken

{ "title": "t2", "author": "a2", "description": "d2" }

The screenshot shows the Postman application interface. At the top, there is a header bar with 'PUT' selected, the URL 'https://baas.kinvey.com/appdata/kid\_rkcLxcUr/books/58340d708ea148fe4a186c39', and various buttons for 'Params', 'Send', 'Save', and 'Code'. Below the header, there are tabs for 'Authorization', 'Headers (2)', 'Body', and 'Pre-request Script'. The 'Body' tab is active, indicated by a blue dot. Under 'Body', there are options for 'form-data', 'x-www-form-urlencoded', 'raw', and 'binary', with 'raw' selected and 'JSON (application/json)' chosen as the type. The raw JSON payload is displayed in a code editor:

```
1 { "title": "t2", "author": "a2", "description": "d2" }
```

At the bottom of the interface, there is a summary section showing 'Status: 200 OK' and 'Time: 223 ms'. Below this, there are tabs for 'Body', 'Cookies', 'Headers (12)', and 'Tests (0/6)'. The 'Body' tab is active. It contains buttons for 'Pretty', 'Raw', 'Preview', and 'JSON'. The JSON response is displayed in a code editor:

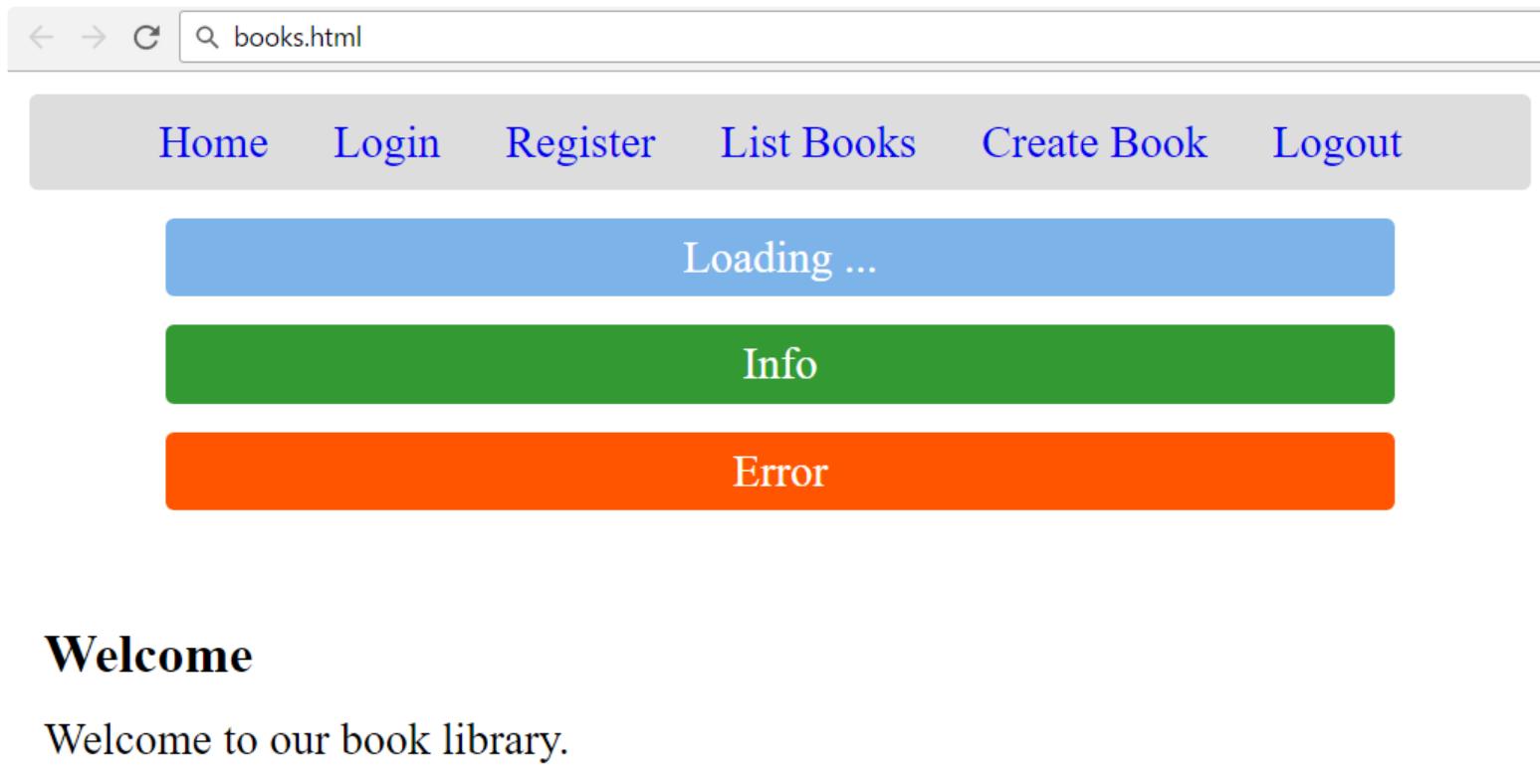
```
1 {
2   "title": "t2",
3   "author": "a2",
4   "description": "d2",
5   "_id": "58340d708ea148fe4a186c39",
6   "_acl": {
7     "creator": "576911067f36dafb32bb60d1"
8   },
9 }
```

# Test the Kinvey Back-End: Delete Book

**DELETE** [https://baas.kinvey.com/appdata/{app\\_id}/books/{id}](https://baas.kinvey.com/appdata/{app_id}/books/{id})

**Authorization:** Kinvey authtoken

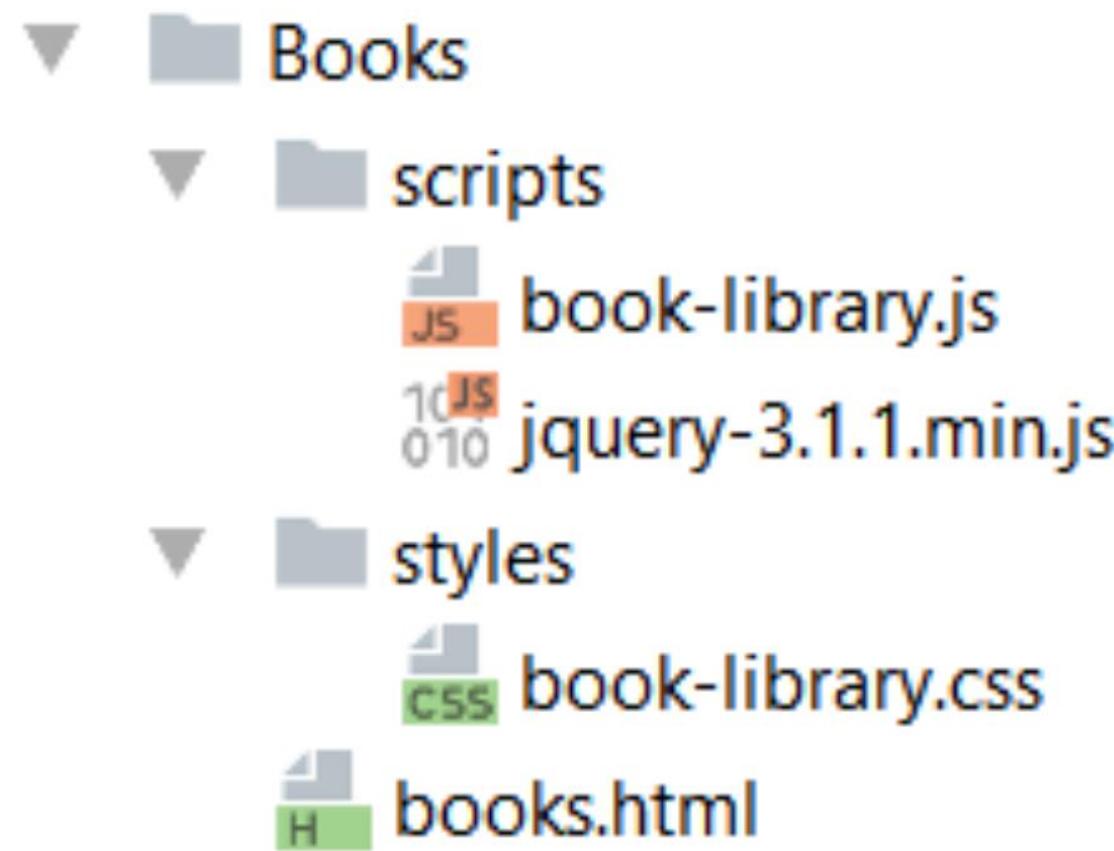
The screenshot shows the Postman application interface. At the top, there is a header bar with a 'DELETE' dropdown, the URL 'https://baas.kinvey.com/appdata/kid\_rkcLxcUr/books/58340d708ea148fe4a186c39', a 'Params' button, a 'Send' button, and a 'Save' button. Below the header, there are tabs for 'Authorization', 'Headers (1)', 'Body', 'Pre-request Script', and 'Tests'. The 'Headers (1)' tab is selected, showing a single entry for 'Authorization' with 'key' and 'value' fields. The 'value' field contains the token 'Kinvey dcf2f66a-48f8-4e50-8db2-fbfee6c1e2cb.+2MmXtaH1SdR7ey'. Below this, the 'Body' tab is selected, showing a JSON response with three lines of code: '1 {', '2 "count": 1', and '3 }'. To the right of the body, status information is displayed: 'Status: 200 OK' and 'Time: 733 ms'. There are also buttons for 'Pretty', 'Raw', 'Preview', 'JSON', 'Save Response', and a magnifying glass icon.



# Създайте скелета на приложението

HTML, CSS, Views, Forms, Info Boxes

# Създайте структура на проекта

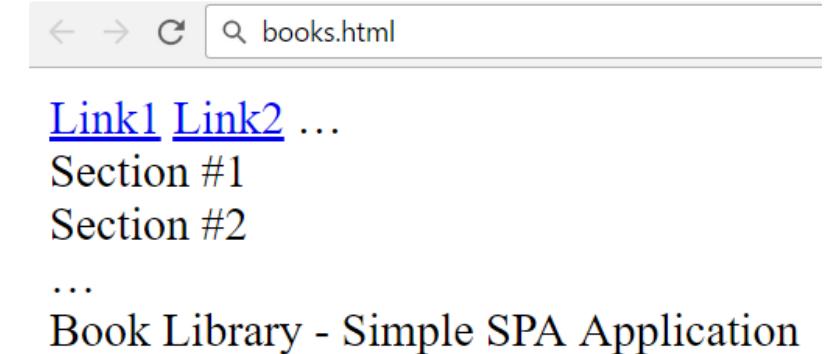


# Започнете с HTML страницата : books.html

```
<!DOCTYPE html>
<html>
  <head>
    <title>Book Library</title>
    <script src="scripts/jquery-3.1.1.min.js"></script>
    <script src="scripts/book-library.js"></script>
    <link rel="stylesheet" type="text/css"
          href="styles/book-library.css" />
  </head>
  <body>...</body>
</html>
```

# HTML структура на тялото

```
<body onload="startApp()">
  <header id="menu">
    <a href="#" id="link1">Link1</a>
    <a href="#" id="link2">Link2</a>
    ...
  </header>
  <main>
    <section id="view1">Section #1</section>
    <section id="view2">Section #2</section>
    ...
  </main>
  <footer>Book Library - Simple SPA Application</footer>
</body>
```



# Основна навигация (Меню)

```
<header id="menu">
  <a href="#" id="linkHome">Home</a>
  <a href="#" id="linkLogin">Login</a>
  <a href="#" id="linkRegister">Register</a>
  <a href="#" id="linkListBooks">List Books</a>
  <a href="#" id="linkCreateBook">Create Book</a>
  <a href="#" id="linkLogout">Logout</a>
  <span id="loggedInUser"></span>
</header>
```



# Раздели на приложения

```
<section id="loadingBox">Loading ...</section>
```

```
<section id="infoBox">Info</section>
```

```
<section id="errorBox">Error</section>
```

```
<section id="viewHome">
  <h1>Welcome</h1>
  Welcome to our book library.
</section>
```

Loading ...  
Info  
Error

Welcome

Welcome to our book library.  
Book Library - Simple SPA Application

## Изглед за вход

```
<section id="viewLogin">
  <h1>Please login</h1>
  <form id="formLogin">
    <div>Username:</div>
    <div><input type="text" name="username" required /></div>
    <div>Password:</div>
    <div><input type="password" name="passwd" required /></div>
    <div><input type="submit" value="Login" /></div>
  </form>
</section>
```

Please login

Username:

Password:

Login

## Изглед за регистрация

```
<section id="viewRegister">
  <h1>Please register here</h1>
  <form id="formRegister">
    <div>Username:</div>
    <div><input type="text" name="username" required /></div>
    <div>Password:</div>
    <div><input type="password" name="passwd" required /></div>
    <div><input type="submit" value="Register" /></div>
  </form>
</section>
```

Please register here

Username:

Password:

Register

# Изглед за книгите

```
<section id="viewBooks">  
  <h1>Books</h1>  
  <div id="books">  
    <table>  
      <tr>  
        <th>Title</th>  
        <th>Author</th>  
        <th>Description</th>  
        <th>Actions</th>  
      </tr>  
      <tr>  
        <td>Book title</td>  
        <td>Book author</td>  
        <td>Book description</td>  
        <td>  
          <a href="#">[Delete]</a>  
          <a href="#">[Edit]</a>  
        </td>  
      </tr>  
      ...  
    </table>  
  </div>  
</section>
```

## Books

Title	Author	Description	Actions
Book title	Book author	Book description	<a href="#">[Delete]</a> <a href="#">[Edit]</a>
Book title	Book author	Book description	<a href="#">[Delete]</a> <a href="#">[Edit]</a>

# Изглед за създаване на книга

```
<section id="viewCreateBook">
  <h1>Create new book</h1>
  <form id="formCreateBook">
    <div>Title:</div>
    <div><input type="text" name="title" required /></div>
    <div>Author:</div>
    <div><input type="text" name="author" required /></div>
    <div>Description:</div>
    <div><textarea name="descr"
      rows="10" required></textarea></div>
    <div><input type="submit" value="Create" /></div>
  </form>
</section>
```

Create new book

Title:

Author:

Description:

Create

# Изглед за редактиране на книга

```
<section id="viewEditBook">
  <h1>Edit existing book</h1>
  <form id="formEditBook">
    <div><input type="hidden" name="id" required /></div>
    <div>Title:</div>
    <div><input type="text" name="title" required /></div>
    <div>Author:</div>
    <div><input type="text" name="author" required /></div>
    <div>Description:</div>
    <div><textarea name="descr" rows="10"
      required></textarea></div>
    <div><input type="submit" value="Edit" /></div>
  </form>
</section>
```

Edit existing book

Title:

Author:

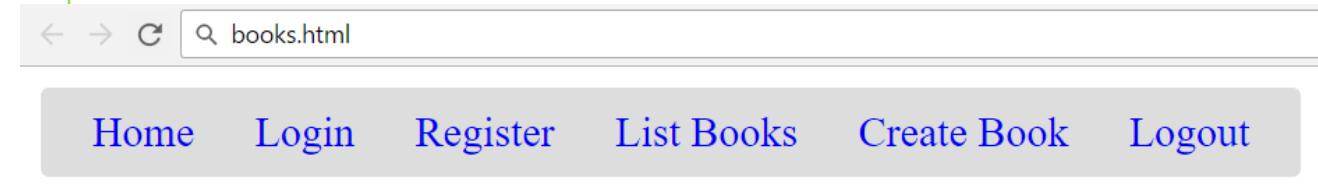
Description:

Edit

# CSS: Оформете лентата за навигация (меню)

```
#menu {  
    background: #DDD;  
    text-align: center;  
    padding: 5px;  
    line-height: 1.5;  
    border-radius: 3px;  
    overflow: auto;  
}  
  
#menu>#loggedInUser {  
    float: right;  
    margin-right: 10px;  
}
```

```
#menu a {  
    text-decoration: none;  
    padding: 5px 10px;  
    border-radius: 5px;  
}  
  
#menu a:hover {  
    background: #BBB;  
}
```



## Оформете секциите и таблициите

```
main > section {  
    display: none;  
    padding: 20px 5px;  
}
```

```
section h1 {  
    margin: 10px 0px;  
    font-size: 1.2em;  
}
```

```
table th {  
    background: #DDD;  
    padding: 10px;  
}  
  
table td {  
    padding: 5px 10px;  
    background: #EEE;  
}
```

### Books

Title	Author	Description	Actions
Book title	Book author	Book description	<a href="#">[Delete]</a> <a href="#">[Edit]</a>
Book title	Book author	Book description	<a href="#">[Delete]</a> <a href="#">[Edit]</a>

## Оформете полетата за зареждане / информация / грешка

```
#infoBox, #errorBox,  
#loadingBox {  
    width: 80%;  
    margin: 10px auto;  
    color: white;  
    text-align: center;  
    padding: 5px;  
    border-radius: 3px;  
}
```



```
#loadingBox {  
    background: #7CB3E9;  
}  
#infoBox {  
    background: #393;  
}  
#errorBox {  
    background: #F50;  
}
```

# Оформете долния колонтиул на приложението

```
footer {  
    background: #DDD;  
    padding: 5px 10px;  
    font-size: 0.8em;  
    text-align: center;  
    border-radius: 3px;  
}
```

Book Library - Simple SPA Application

# Тествайте скелета на приложението

A screenshot of a web browser window titled "books.html". The address bar shows the URL. Below the title bar is a navigation menu with links: Home, Login, Register, List Books, Create Book, and Logout. A large blue horizontal bar spans most of the page width, containing the text "Loading ...". Below this is a green horizontal bar labeled "Info", and further down is an orange horizontal bar labeled "Error".

## Welcome

Welcome to our book library.

## Please login

Username:

guest

Password:

.....

Login

A screenshot of a web browser window titled "books.html". The address bar shows the URL. Below the title bar is a header section with the word "Books". Underneath is a table with four columns: Title, Author, Description, and Actions. Each row contains two entries, each with a "Delete" and an "Edit" link. The table has a light gray background with alternating row colors.

Title	Author	Description	Actions
Book title	Book author	Book description	<a href="#">[Delete]</a> <a href="#">[Edit]</a>
Book title	Book author	Book description	<a href="#">[Delete]</a> <a href="#">[Edit]</a>

## Create new book

Title:

Author:

Description:

Create

Screenshot of the Google Chrome DevTools Application tab showing local storage data.

The Application tab is selected, displaying the following table:

	Key	Value
authToken	332f69d0-c19e-4f6c-acc8-4190b08e6bec.oM786OBFp4q5K7QGpAKz1qLaMQqmDJI...	
userId	576911067f36dafb32bb60d1	

The left sidebar shows the Storage section with Local Storage and Session Storage expanded. The Session Storage for 'http://localhost' is selected, indicated by a blue background.

Icons representing different storage types are displayed below the table:

- Local Storage: A globe icon.
- Session Storage: A database cylinder icon with 'ID' and a signature-like 'm'.
- IndexedDB: A filing cabinet icon.
- Web SQL: A yellow square with the letters 'JS'.

# Session / Local Storage

# Session / Local Storage - Overview

- ▶ Съхранението на сесията съдържа двойки ключ / стойност в сесията на браузъра
- ▶ Всички данни се губят, когато браузърът е затворен, оцелява при презареждане на страници

```
// Save data to sessionStorage  
sessionStorage.setItem('username', 'maria');
```

```
// Get saved data from sessionStorage  
let currentUser = sessionStorage.getItem('username');
```

```
// Remove all saved data from sessionStorage  
sessionStorage.clear();
```

# Local Storage

- ▶ Локалното хранилище съдържа двойки ключ / стойност в браузъра
- ▶ Данните оцеляват дълго време, докато не бъдат изтрити ръчно
- ▶ Всеки произход (местоположение на сайта) има собствено хранилище
- ▶ <https://facebook.com> съдържа данни, различни от <https://google.com>

```
// Save data to localStorage
localStorage.setItem('language', 'en');
```

```
// Get saved data from localStorage
let lang = localStorage.getItem('language');
```

The screenshot shows the Google Chrome DevTools interface with the "Application" tab selected. On the left, a sidebar titled "Storage" lists "Local Storage" and "Session Storage". Under "Session Storage" for the domain "http://localhost", there is one item: "username" with the value "todor". Below the sidebar is a table with two columns: "Key" and "Value". At the bottom of the sidebar, there are refresh and close buttons. The main area contains a "Console" tab and a log history. The log shows the command `sessionStorage.setItem("username", "todor");` followed by its return value `undefined`. There are also icons for filtering the log.

Storage

▶ Local Storage

▼ Session Storage

http://localhost

Key Value

username todor

Console

Preserve log

```
> sessionStorage.setItem("username", "todor");
< undefined
```

# Структура на приложението

## book-library.js

```
function startApp() {  
    sessionStorage.clear(); // Clear user auth data  
    showHideMenuLinks();  
    showView('viewHome');  
    // Bind the navigation menu links  
    $("#linkHome").click(showHomeView);  
    ...  
    // Bind the form submit actions  
    $("#formLogin").submit(loginUser);  
    ...  
    $("form").submit(function(e) { e.preventDefault() });  
}
```

```
// Bind the navigation menu links
$("#linkHome").click(showHomeView);
$("#linkLogin").click(showLoginView);
$("#linkRegister").click(showRegisterView);
$("#linkListBooks").click(listBooks);
$("#linkCreateBook").click(showCreateBookView);
$("#linkLogout").click(logoutUser);
```

```
// Bind the form submit buttons
$("#buttonLoginUser").click(loginUser);
$("#buttonRegisterUser").click(registerUser);
$("#buttonCreateBook").click(createBook);
$("#buttonEditBook").click(editBook);
```

```
// Bind the info / error boxes: hide on click
$("#infoBox, #errorBox").click(function() {
  $(this).fadeOut();
});
```

```
// Attach AJAX "loading" event listener
$(document).on({
  ajaxStart: function() { $("#loadingBox").show() },
  ajaxStop: function() { $("#loadingBox").hide() }
});
```

```
function showHideMenuLinks() {  
    $("#linkHome").show();  
    if (sessionStorage.getItem('authToken')) {  
        // We have logged in user  
        $("#linkLogin").hide();  
        $("#linkRegister").hide();  
        $("#linkListBooks").show();  
        $("#linkCreateBook").show();  
        $("#linkLogout").show();  
    } else {  
        $("#linkLogout").hide();  
    }  
}
```

```
    } else {
        // No logged in user
        $("#linkLogin").show();
        $("#linkRegister").show();
        $("#linkListBooks").hide();
        $("#linkCreateBook").hide();
        $("#linkLogout").hide();
    }
}
```

```
function showView(viewName) {  
    // Hide all views and show the selected view only  
    $('main > section').hide();  
    $('#' + viewName).show();  
}
```

```
function showHomeView() {  
    showView('viewHome');  
}
```

```
function showLoginView() {  
    showView('viewLogin');  
    $('#formLogin').trigger('reset');  
}
```

```
function showRegisterView() {  
    $('#formRegister').trigger('reset');  
    showView('viewRegister');  
}
```

```
function showCreateBookView() {  
    $('#formCreateBook').trigger('reset');  
    showView('viewCreateBook');  
}
```

```
function loginUser() { // TODO }
```

```
function registerUser() { // TODO }
```

```
function logoutUser() { // TODO }
```

```
function listBooks() {
  // TODO: to be implemented later
}
```

```
function createBook() { // TODO }
```

```
function editBook() { // TODO }
```

```
function deleteBook() { // TODO }
```

# Тествайте навигацията в приложението

The image displays three separate browser windows, each showing a different state of a Single Page Application (SPA) with the URL "books.html" in the address bar.

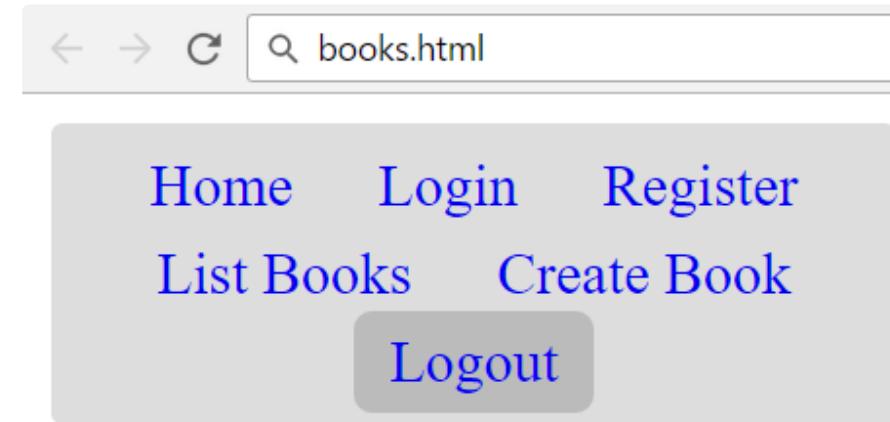
- Left Window:** Shows the "Home" page. The title bar says "books.html". The header contains "Home", "Login", and "Register" buttons. The main content area displays "Welcome" and "Welcome to our book library.". A footer bar at the bottom says "Book Library - Simple SPA Application".
- Middle Window:** Shows the "Login" page. The title bar says "books.html". The header contains "Home", "Login" (which is highlighted), and "Register" buttons. The main content area displays "Please login" followed by "Username:" and "Password:" fields, and a "Login" button. A footer bar at the bottom says "Book Library - Simple SPA Application".
- Right Window:** Shows the "Register" page. The title bar says "books.html". The header contains "Home", "Login", and "Register" buttons. The main content area displays "Please register here" followed by "Username:" and "Password:" fields, and a "Register" button. A footer bar at the bottom says "Book Library - Simple SPA Application".

# Please login

Username:

Password:

Login



# Please register here

Username:

Password:

Register

# Вход / Регистрация / Изход

User Management with Kinvey

# App Constants

```
const kinveyBaseUrl = "https://baas.kinvey.com/";
const kinveyAppKey = "kid_rkcLxcUr";
const kinveyAppSecret =
  "e234a245b3864b2eb7ee41e19b8ca4e5";
const kinveyAppAuthHeaders = {
  'Authorization': "Basic " +
    btoa(kinveyAppKey + ":" + kinveyAppSecret),
};
```

## Регистрация на потребител: Заявка AJAX

```
function registerUser() {  
    let userData = {  
        username: $('#formRegister input[name=username]').val(),  
        password: $('#formRegister input[name=passwd]').val()  
    };  
    $.ajax({  
        method: "POST",  
        url: kinveyBaseUrl + "user/" + kinveyAppKey + "/",  
        headers: kinveyAppAuthHeaders,  
        data: userData,  
        success: registerSuccess,  
        error: handleAjaxError  
    });  
}
```

## Регистрация на потребител: След заявка за AJAX

```
function registerUser() {  
    ...  
  
    function registerSuccess(userInfo) {  
        saveAuthInSession(userInfo);  
        showHideMenuLinks();  
        listBooks();  
        showInfo('User registration successful.');//  
    }  
}
```

## Запомните данните за удостоверяване на потребителя

```
function saveAuthInSession(userInfo) {  
    let userAuth = userInfo._kmd.authToken;  
    sessionStorage.setItem('authToken', userAuth);  
    let userId = userInfo._id;  
    sessionStorage.setItem('userId', userId);  
    let username = userInfo.username;  
    $('#loggedInUser').text(  
        "Welcome, " + username + "!");  
}
```

## Обработка на AJAX грешки : Покажете полето за грешка

```
function handleAjaxError(response) {  
    let errorMsg = JSON.stringify(response);  
    if (response.readyState === 0)  
        errorMsg = "Cannot connect due to network  
error."  
    if (response.responseJSON &&  
        response.responseJSON.description)  
        errorMsg = response.responseJSON.description;  
    showError(errorMsg);  
}
```

## Показване на информация / съобщение за грешка

```
function showInfo(message) {  
    $('#infoBox').text(message);  
    $('#infoBox').show();  
    setTimeout(function() {  
        $('#infoBox').fadeOut();  
    }, 3000);  
}
```

```
function showError(errorMsg) {  
    $('#errorBox').text("Error: " + errorMsg);  
    $('#errorBox').show();  
}
```

[Home](#)[List Books](#)[Create Book](#)[Logout](#)

Welcome, stef!

User registration successful.

## Please register here

Username:

Password:

Register

Home Login Register

Error: This username is already taken. Please  
retry your request with a different username

## Please register here

Username:

  
stef

Password:

  
.....

Register

```
function loginUser() {  
    let userData = {  
        username: $('#formLogin input[name=username]').val(),  
        password: $('#formLogin input[name=passwd]').val()  
    };  
    $.ajax({  
        method: "POST",  
        url: kinveyBaseUrl + "user/" + kinveyAppKey + "/login",  
        headers: kinveyAppAuthHeaders,  
        data: userData,  
        success: loginSuccess,  
        error: handleAjaxError  
    });  
}
```

```
function loginUser() {  
  ...  
  
  function loginSuccess(userInfo) {  
    saveAuthInSession(userInfo);  
    showHideMenuLinks();  
    listBooks();  
    showInfo('Login successful.');//  
  }  
}
```

```
function logoutUser() {  
    sessionStorage.clear();  
    $('#loggedInUser').text("");  
    showHideMenuLinks();  
    showView('viewHome');  
    showInfo('Logout successful.');//  
}
```



# Прилагане на CRUD операции

List / Create / Delete / Edit

## Списък на книгите: Заявка AJAX

```
function listBooks() {
    $('#books').empty();
    showView('viewBooks');

    $.ajax({
        method: "GET",
        url: kinveyBaseUrl + "appdata/" + kinveyAppKey + "/books",
        headers: getKinveyUserAuthHeaders(),
        success: loadBooksSuccess,
        error: handleAjaxError
    });

    function loadBooksSuccess(books) { ... }
}
```

# Kinvey Authorization Headers

```
function getKinveyUserAuthHeaders() {  
    return {  
        'Authorization': "Kinvey " +  
            sessionStorage.getItem('authToken'),  
    };  
}
```

## Списък на книгите: След заявка за AJAX

```
function loadBooksSuccess(books) {
    showInfo('Books loaded.');
    if (books.length == 0) {
        $('#books').text('No books in the library.');
    } else {
        let booksTable = $('<table>')
            .append('<tr>').append(
                '<th>Title</th><th>Author</th>',
                '<th>Description</th><th>Actions</th>'));
        for (let book of books)
            appendBookRow(book, booksTable);
        $('#books').append(booksTable);
    }
}
```

## Показване на единична книга

```
function appendBookRow(book, booksTable) {  
    let links = [];  
    // TODO: action links will come later  
  
    booksTable.append($('|').append(  
        $(' ').text(book.title),         $(' ').text(book.author),         $(' ').text(book.description),         $(' ').append(links)     )); } | | | |

```

# Тест: Списък на книгите

The screenshot shows a web browser window with the URL 'books.html' in the address bar. The page title is 'Books'. The header includes links for 'Home', 'List Books', 'Create Book', and 'Logout', along with a welcome message 'Welcome, guest!'. The main content area displays a table of books with columns for Title, Author, Description, and Actions. The table contains four rows of data.

Title	Author	Description	Actions
Java Fundamentals	Bay Ivan	Strong Java book. Not for beginners	
Software Technologies	Some author  	Creating simple apps with HTML5, PHP, C# and Java.	
C# Basics	Some author  	Very good C# book!	
Some title  	Some author  	Some description  	<a href="#">[Delete]</a> <a href="#">[Edit]</a>

Book Library - Simple SPA Application

# Създайте нова книга: Заявка AJAX

```
function createBook() {  
    let bookData = {  
        title: $('#formCreateBook input[name=title]').val(),  
        author: $('#formCreateBook input[name=author]').val(),  
        description: $('#formCreateBook textarea[name=descr]').val()  
    };  
    $.ajax({  
        method: "POST",  
        url: kinveyBaseUrl + "appdata/" + kinveyAppKey + "/books",  
        headers: getKinveyUserAuthHeaders(),  
        data: bookData,  
        success: createBookSuccess,  
        error: handleAjaxError  
    });  
}
```

## Създаване на нова книга: След заявка AJAX

```
function createBookSuccess(response) {  
    listBooks();  
    showInfo('Book created.');//  
}  
};
```

# Показване на Редактиране / Изтриване на връзки

```
function appendBookRow(book, booksTable) {
  let links = [];
  if (book._acl.creator == sessionStorage['userId']) {
    let deleteLink = $('<a href="#">[Delete]</a>')
      .click(deleteBook.bind(this, book));
    let editLink = $('<a href="#">[Edit]</a>')
      .click(loadBookForEdit.bind(this, book));
    links = [deleteLink, ' ', editLink];
  }
  booksTable.append($('<tr>')
    .append( ... cells & links ));
}
```

# Изтриване на книга: Заявка AJAX

```
function deleteBook(book) {
    $.ajax({
        method: "DELETE",
        url: kinveyBookUrl = kinveyBaseUrl + "appdata/" +
            kinveyAppKey + "/books/" + book._id,
        headers: getKinveyUserAuthHeaders(),
        success: deleteBookSuccess,
        error: handleAjaxError
    });

    function deleteBookSuccess(response) {
        listBooks();
        showInfo('Book deleted.');
    }
}
```

# Тест: Изтриване на книга



Book deleted.

A screenshot of the Network tab in a browser's developer tools. The tab is active, and the main content area shows a list of network requests. One request is selected, showing detailed information:

Name	Headers	Preview	Response	Timing
583322da6bff1d206...	▼ General			
583322da6bff1d206...	<b>Request URL:</b> https://baas.kinvey.com/appdata/kid_rkcLxcUr/books/583322da6bff1d20669e5ecc			
books	<b>Request Method:</b> DELETE			
books	<b>Status Code:</b> 200 OK			
	<b>Remote Address:</b> 52.70.173.60:443			

## Зареждане на книга за редактиране: AJAX заявка

```
function loadBookForEdit(book) {  
    $.ajax({  
        method: "GET",  
        url: kinveyBookUrl = kinveyBaseUrl + "appdata/" +  
            kinveyAppKey + "/books/" + book._id,  
        headers: getKinveyUserAuthHeaders(),  
        success: loadBookForEditSuccess,  
        error: handleAjaxError  
    });  
    ...  
}
```

## Зареждане на книга за редактиране: След заявка AJAX

```
function loadBookForEditSuccess(book) {  
    $('#formEditBook input[name=id]').val(book._id);  
    $('#formEditBook input[name=title]').val(book.title);  
    $('#formEditBook input[name=author]')  
        .val(book.author);  
    $('#formEditBook textarea[name=descr]')  
        .val(book.description);  
    showView('viewEditBook');  
}  
}
```

## Редактиране на книга: Заявка AJAX

```
function editBook() {  
    let bookData = {  
        title: $('#formEditBook input[name=title]').val(),  
        author: $('#formEditBook input[name=author]').val(),  
        description:  
            $('#formEditBook textarea[name=descr]').val()  
    };  
    $.ajax({  
        method: "PUT",  
        url: kinveyBaseUrl + "appdata/" + kinveyAppKey +  
            "/books/" + $('#formEditBook input[name=id]').val(),  
        data: bookData  
    }).done(function(response) {  
        console.log("Book updated successfully: " + response._id);  
    }).fail(function(error) {  
        console.error("Error updating book: " + error.message);  
    });  
}
```

## Редактиране на книга: Заявка AJAX (2)

```
headers: getKinveyUserAuthHeaders(),
data: bookData,
success: editBookSuccess,
error: handleAjaxError
});

function editBookSuccess(response) {
    listBooks();
    showInfo('Book edited.');
}

}
```

## Резюме:

- ▶ Едностранично приложение (SPA) изградено чрез HTML5, AJAX и REST + back-end
- ▶ Навигацията в приложението се състои от DOM елементи, които се показват / скриват
- ▶ Входът / регистрацията / излизането обикновено се реализира с sessionStorage
- ▶ CRUD операциите изпращат AJAX заявка и представят резултатите след това
- ▶ При редактирането / изтриването се налага първо да заредите елемента, след това да го редактирате / потвърдите изтриването, след което да публикувате промени

