

REST API + HATEOAS biblioteki.

Modele: Autor, Książka, Wypożyczenie, Rezerwacja, Użytkownik

Baza danych: h2, hasła haszowane za pomocą BCrypta

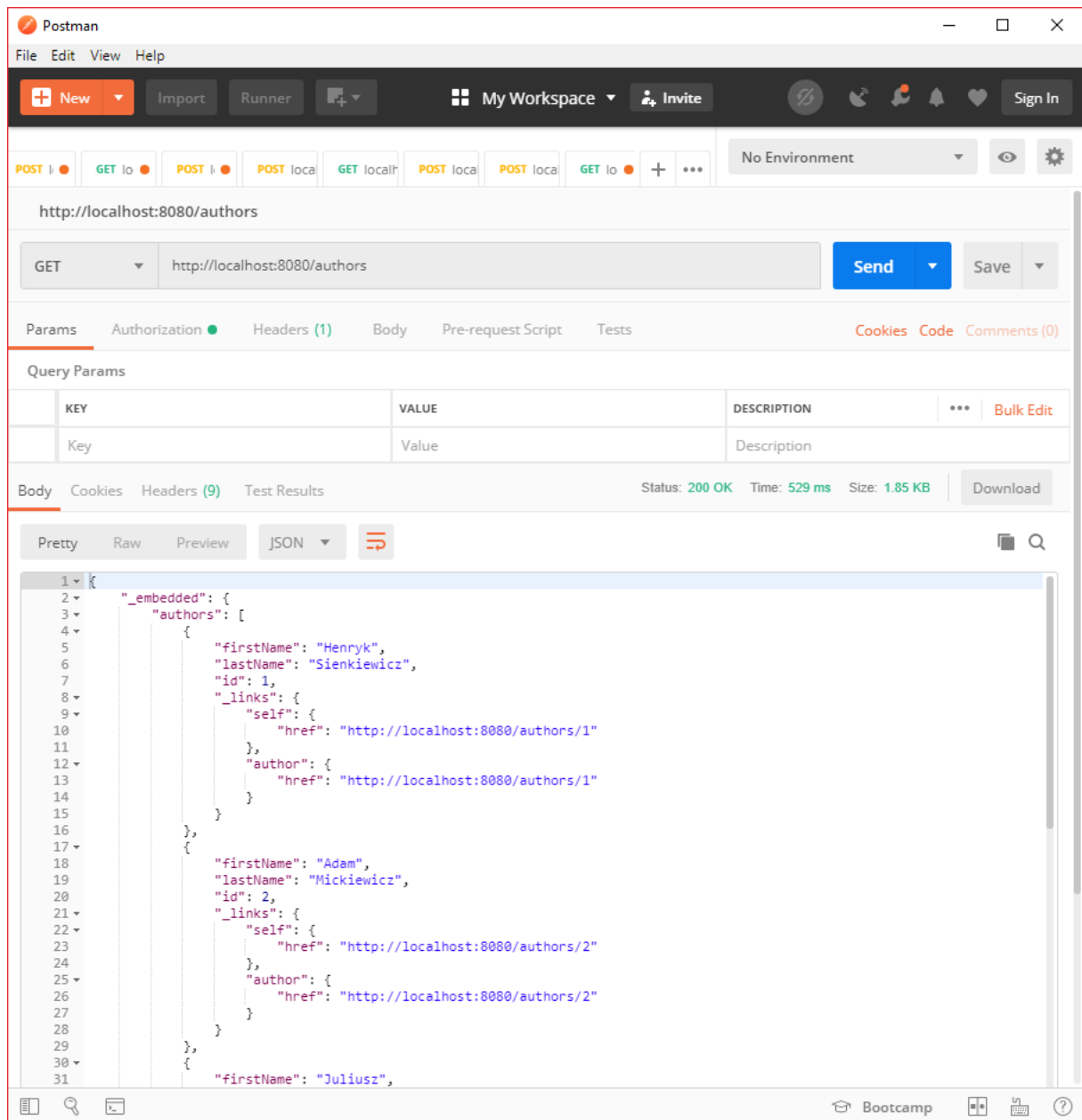
Dwa poziomy uprawnień: Admin i User

Metoda prezentacji: program Postman

Link do githuba: [https://github.com/ArturGrzegorzKancierz/Projekt\\_Zaliczeniowy](https://github.com/ArturGrzegorzKancierz/Projekt_Zaliczeniowy)

Skrypt wypełniający bazę w pliku data.sql w katalogu Resources

Authors:



Postman interface showing a GET request to `http://localhost:8080/authors`. The response is a 200 OK status with a time of 529 ms and a size of 1.85 KB. The response body is displayed in JSON format, showing a list of authors with their first names, last names, IDs, and self links.

```
{
  "_embedded": {
    "authors": [
      {
        "firstName": "Henryk",
        "lastName": "Sienkiewicz",
        "id": 1,
        "_links": {
          "self": {
            "href": "http://localhost:8080/authors/1"
          },
          "author": {
            "href": "http://localhost:8080/authors/1"
          }
        }
      },
      {
        "firstName": "Adam",
        "lastName": "Mickiewicz",
        "id": 2,
        "_links": {
          "self": {
            "href": "http://localhost:8080/authors/2"
          },
          "author": {
            "href": "http://localhost:8080/authors/2"
          }
        }
      },
      {
        "firstName": "Juliusz",
```

GET: authors/id

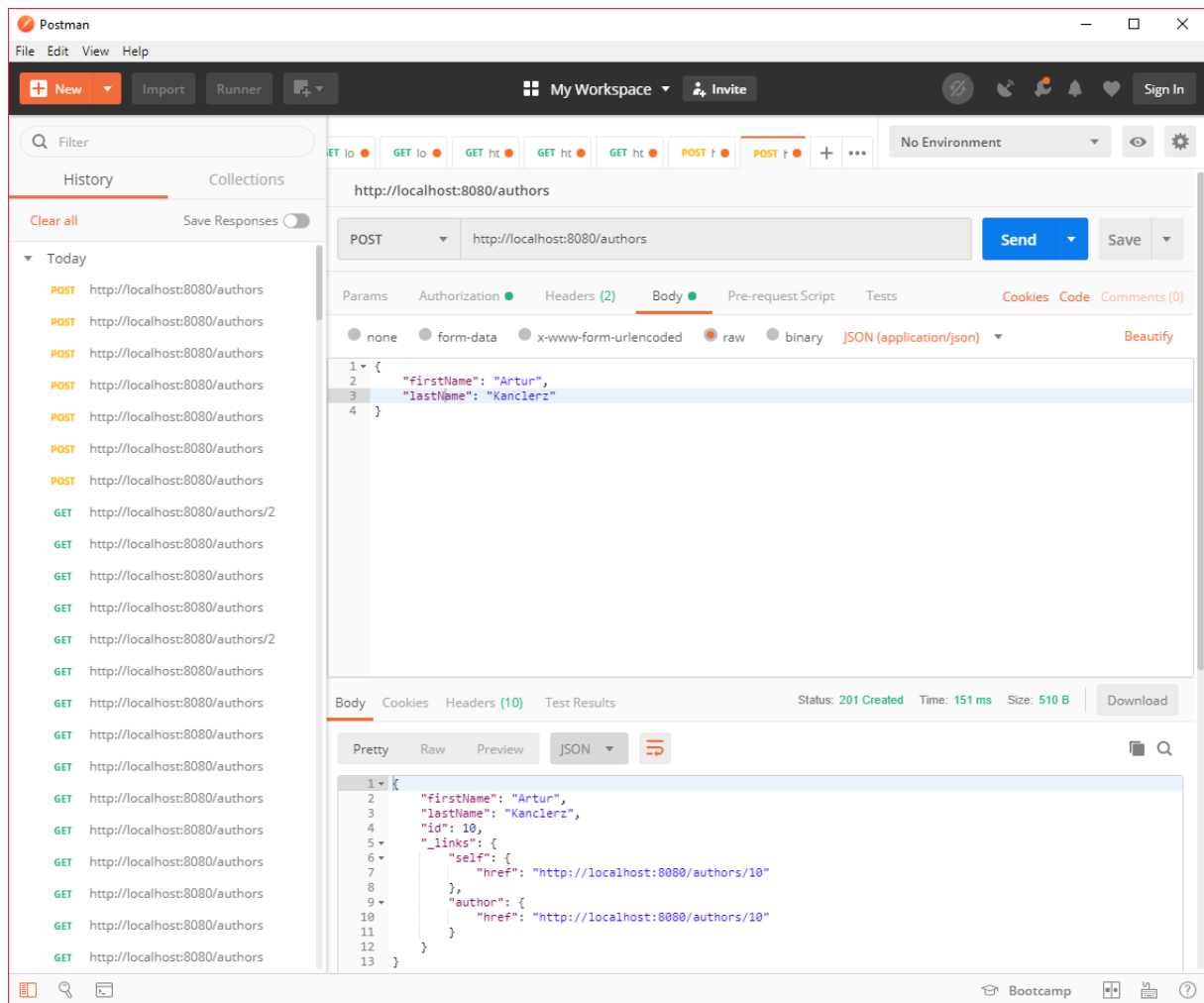
The screenshot shows the Postman application interface. The top bar includes the Postman logo, menu items (File, Edit, View, Help), and workspace controls (My Workspace, Invite). The left sidebar shows a list of requests under the 'History' tab, with a filter input and 'Clear all' and 'Save Responses' buttons. The main panel displays a GET request to `http://localhost:8080/authors/2`. The request is configured with the following details:

- Method:** GET
- URL:** `http://localhost:8080/authors/2`
- Params:** None
- Headers:** 1 header (Key, Value, Description)
- Body:** None
- Test Results:** Status: 200 OK, Time: 119 ms, Size: 480 B

The response body is displayed in the 'Body' tab, showing a JSON object in 'Pretty' format:

```
1 {
2   "firstName": "Adam",
3   "lastName": "Mickiewicz",
4   "id": 2,
5   "_links": {
6     "self": {
7       "href": "http://localhost:8080/authors/2"
8     },
9     "author": {
10      "href": "http://localhost:8080/authors/2"
11    }
12  }
13 }
```

## POST: authors



## GET: Books

The image shows the Postman application interface. On the left, the 'History' tab is active, displaying a list of recent requests. The main workspace shows a GET request to `http://localhost:8080/books`. The response is a JSON object, displayed in the 'Body' tab. The response status is 200 OK, with a time of 299 ms and a size of 3.15 KB. The JSON response is as follows:

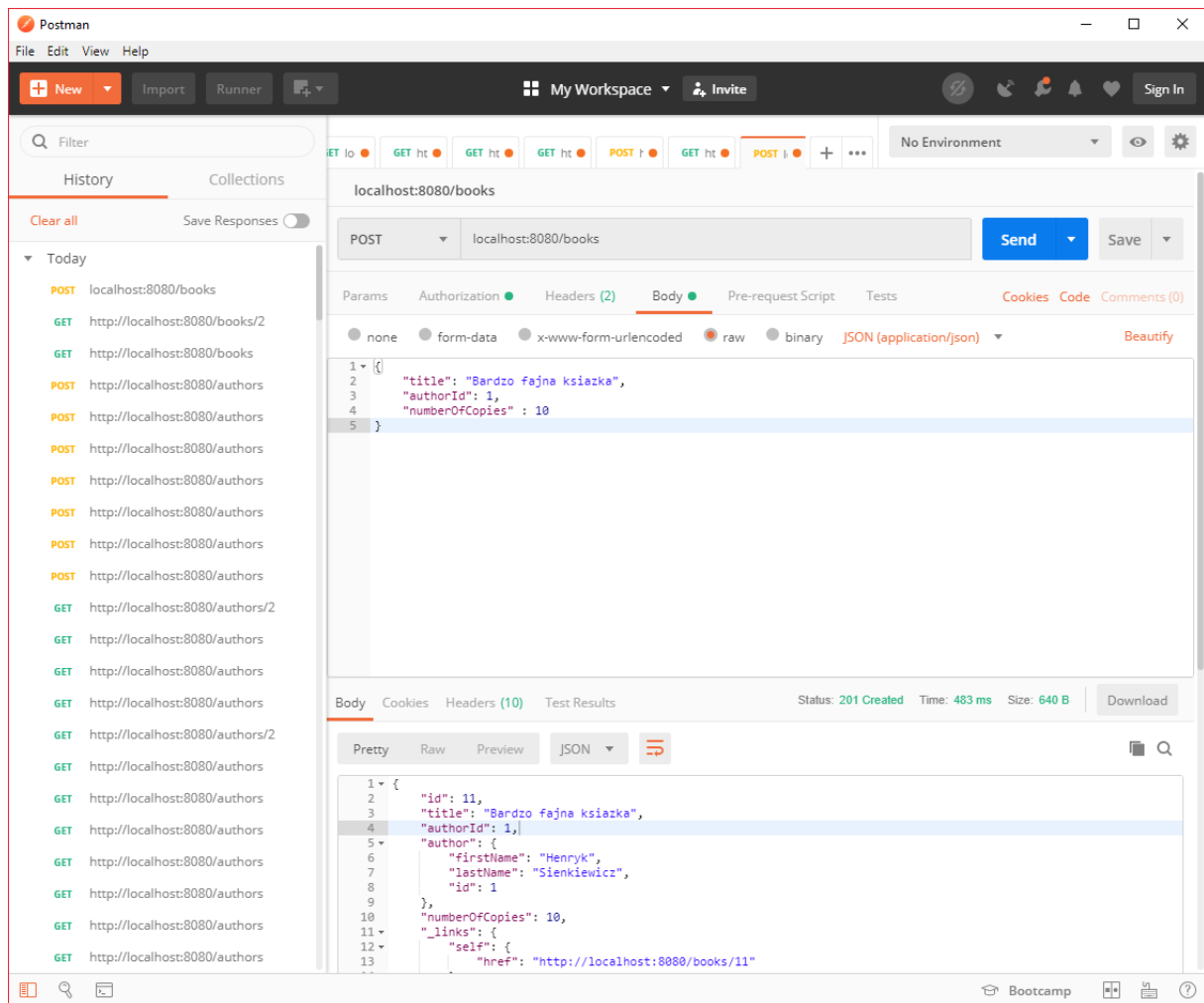
```
{
  "_embedded": {
    "books": [
      {
        "id": 1,
        "title": "Pan Tadeusz",
        "authorId": 2,
        "author": {
          "firstName": "Adam",
          "lastName": "Mickiewicz",
          "id": 2
        },
        "numberOfCopies": 7,
        "_links": {
          "self": {
            "href": "http://localhost:8080/books/1"
          },
          "book": {
            "href": "http://localhost:8080/books/1"
          },
          "author": {
            "href": "http://localhost:8080/authors/2"
          }
        }
      }
    ]
  }
}
```

## GET: books/id

The image shows the Postman application interface. The top bar includes the Postman logo, menu options (File, Edit, View, Help), and workspace management tools (New, Import, Runner, My Workspace, Invite). The main area is divided into a left sidebar and a right pane. The sidebar shows a 'History' tab with a list of recent requests, including GET requests to `http://localhost:8080/books/2` and `http://localhost:8080/books`, and POST requests to `http://localhost:8080/authors`. The right pane displays the details of a selected GET request to `http://localhost:8080/books/2`. The request is configured with the method 'GET' and the URL `http://localhost:8080/books/2`. The response is shown in the 'Body' tab, displaying a JSON object in 'Pretty' format. The status is '200 OK', the time is '120 ms', and the size is '599 B'. The JSON response is as follows:

```
1 {
2   "id": 2,
3   "title": "Quo Vadis",
4   "authorId": 1,
5   "author": {
6     "firstName": "Henryk",
7     "lastName": "Sienkiewicz",
8     "id": 1
9   },
10  "numberOfCopies": 4,
11  "_links": {
12    "self": {
13      "href": "http://localhost:8080/books/2"
14    },
15    "book": {
16      "href": "http://localhost:8080/books/2"
17    },
18    "author": {
19      "href": "http://localhost:8080/authors/1"
20    }
21  }
22 }
```

POST: books/



GET: borrows/

The image shows the Postman application interface. The top bar includes the Postman logo, menu options (File, Edit, View, Help), and workspace controls (My Workspace, Invite). The left sidebar shows a history of requests, with the current request being a GET to localhost:8080/borrows. The main panel displays the request details for localhost:8080/borrows, including the method (GET), the URL, and the response status (200 OK). The response body is shown in JSON format, displaying a list of borrow records with fields like id, reservationId, username, borrowDate, returnDate, dateOfReturn, and \_links.

localhost:8080/borrows

GET localhost:8080/borrows

Send Save

Params Authorization Headers (2) Body Pre-request Script Tests Cookies Code Comments (0)

none form-data x-www-form-urlencoded raw binary JSON (application/json) Beautify

1

Body Cookies Headers (9) Test Results Status: 200 OK Time: 125 ms Size: 934 B Download

Pretty Raw Preview JSON

```
1 {
2   "_embedded": {
3     "borrows": [
4       {
5         "id": 1,
6         "reservationId": 1,
7         "username": "user",
8         "borrowDate": "2019-03-29T14:22:32.622+0000",
9         "returnDate": null,
10        "dateOfReturn": "2019-04-12T13:22:32.622+0000",
11        "_links": {
12          "self": {
13            "href": "http://localhost:8080/borrows/1"
14          },
15          "borrow": {
16            "href": "http://localhost:8080/borrows/1"
17          }
18        }
19      },
20      {
21        "id": 2,
22        "reservationId": 2,
23        "username": "user",
24        "borrowDate": "2019-03-29T14:23:19.505+0000",
25        "returnDate": null,
26        "dateOfReturn": "2019-04-12T13:23:19.505+0000",
27        "_links": {
28          "self": {
29            "href": "http://localhost:8080/borrows/2"
30          },
31          "borrow": {
```



## GET: borrows/id

The image shows the Postman application interface. The top bar includes the Postman logo, menu options (File, Edit, View, Help), and workspace management tools (New, Import, Runner, My Workspace, Invite). The left sidebar displays a history of requests, with the current selection being a GET request to `localhost:8080/borrows/1`. The main panel shows the details of this request, including the URL, method (GET), and the response body. The response is a JSON object with the following structure:

```
1 {
2   "id": 1,
3   "reservationId": 1,
4   "username": "user",
5   "borrowDate": "2019-03-29T14:22:32.622+0000",
6   "returnDate": null,
7   "dateOfReturn": "2019-04-12T13:22:32.622+0000",
8   "_links": {
9     "self": {
10      "href": "http://localhost:8080/borrows/1"
11    },
12    "borrow": {
13      "href": "http://localhost:8080/borrows/1"
14    }
15  }
16 }
```

The status bar at the bottom indicates a successful response (Status: 200 OK) with a response time of 131 ms and a size of 581 B. The bottom right corner shows the Bootcamp logo and other navigation icons.

PUT: borrows/id (slajd wyżej wydać, że returnDate = null)

The screenshot shows the Postman application interface. On the left, the 'History' tab is active, displaying a list of recent requests. The main workspace shows a PUT request to `localhost:8080/borrows/1`. The request body is a JSON object with the following structure:

```
1 {
2   "returnDate": "2019-03-30T12:08:50.711+0000"
3 }
```

The response status is `200 OK`, with a time of `127 ms` and a size of `608 B`. The response body is displayed in the 'Pretty' view as a JSON object:

```
1 {
2   "id": 1,
3   "reservationId": -1,
4   "username": "user",
5   "borrowDate": "2019-03-29T12:22:32.622+0000",
6   "returnDate": "2019-03-30T12:08:50.711+0000",
7   "dateOfReturn": "2019-04-12T09:22:32.622+0000",
8   "_links": {
9     "self": {
10      "href": "http://localhost:8080/borrows/1"
11    },
12    "borrow": {
13      "href": "http://localhost:8080/borrows/1"
14    }
15  }
16 }
```

## POST: reservations/

The image shows the Postman application interface. On the left, the 'History' tab is active, displaying a list of recent requests. The main workspace shows a POST request to `localhost:8080/reservations/`. The request body is a JSON object with `"bookId": 1`. The response is a 201 Created status with a JSON body containing reservation details.

**Request:**

```
POST localhost:8080/reservations/
{
  "bookId": 1
}
```

**Response:**

```
201 Created
{
  "id": 7,
  "bookId": 1,
  "username": "admin",
  "reservationDate": "2019-07-09T19:22:32.370+0000",
  "_links": {
    "self": {
      "href": "http://localhost:8080/reservations/7"
    },
    "reservation": {
      "href": "http://localhost:8080/reservations/7"
    }
  }
}
```

GET: reservations/

The image shows the Postman application interface. The top bar includes the Postman logo, menu options (File, Edit, View, Help), and workspace controls (My Workspace, Invite). The main area is divided into a left sidebar and a right pane. The sidebar shows a 'History' tab with a list of recent requests, including GET and POST requests to localhost:8080/reservations/ and PUT requests to localhost:8080/borrows/. The right pane displays a GET request to localhost:8080/reservations/. The response is shown in the 'Pretty' tab, displaying a JSON array of three reservation objects. Each object contains fields for id, bookId, username, reservationDate, and a self link. The first reservation has id 7, bookId 1, and reservationDate 2019-07-09T19:22:32.370+0000. The second has id 8, bookId 9, and reservationDate 2019-07-09T19:23:44.852+0000. The third has id 9, bookId 5, and reservationDate 2019-07-09T19:24:39.951+0000.

```
72
73
74 {
75   "id": 7,
76   "bookId": 1,
77   "username": "admin",
78   "reservationDate": "2019-07-09T19:22:32.370+0000",
79   "_links": {
80     "self": {
81       "href": "http://localhost:8080/reservations/7"
82     },
83     "reservation": {
84       "href": "http://localhost:8080/reservations/7"
85     }
86   }
87 },
88 {
89   "id": 8,
90   "bookId": 9,
91   "username": "admin",
92   "reservationDate": "2019-07-09T19:23:44.852+0000",
93   "_links": {
94     "self": {
95       "href": "http://localhost:8080/reservations/8"
96     },
97     "reservation": {
98       "href": "http://localhost:8080/reservations/8"
99     }
100   }
101 },
102 {
103   "id": 9,
104   "bookId": 5,
105   "username": "admin",
106   "reservationDate": "2019-07-09T19:24:39.951+0000",
107   "_links": {
108     "self": {
109       "href": "http://localhost:8080/reservations/9"
110     },
111     "reservation": {
112       "href": "http://localhost:8080/reservations/9"
113     }
114   }
115 },
116 ],
117 "_links": {
118   "self": {
119     "href": "http://localhost:8080/reservations"
120   }
121 }
122 }
123 }
```

PUT: reservations/ (poprzedni slajd – id rezerwacji: 8, id książki: 5)

The screenshot shows the Postman application interface. The top bar includes the Postman logo, menu options (File, Edit, View, Help), and workspace controls (My Workspace, Invite). The left sidebar displays a history of requests, with the current selection being a PUT request to localhost:8080/reservations/9. The main panel shows the details of this request, including the method (PUT), URL (localhost:8080/reservations/9), and the body (JSON). The body content is a JSON object with the following structure:

```
1 {
2   "bookId": 10
3 }
```

The response section at the bottom shows the status (200 OK), time (137 ms), and size (532 B). The response body is displayed in a pretty-printed JSON format:

```
1 {
2   "id": 9,
3   "bookId": 10,
4   "username": "admin",
5   "reservationDate": "2019-07-09T19:26:32.729+0000",
6   "_links": {
7     "self": {
8       "href": "http://localhost:8080/reservations/9"
9     },
10    "reservation": {
11      "href": "http://localhost:8080/reservations/9"
12    }
13  }
14 }
```

GET: users/

The image shows the Postman application interface. The top bar includes the Postman logo, menu options (File, Edit, View, Help), and workspace controls (My Workspace, Invite). The left sidebar shows a history of requests, with the current request being a GET to localhost:8080/users. The main panel displays the request details and the response body in JSON format.

Request Details:

- Method: GET
- URL: localhost:8080/users
- Environment: No Environment

Response Body (JSON):

```
1 {
2   "_embedded": {
3     "users": [
4       {
5         "username": "admin",
6         "email": "admin@admin",
7         "_links": {
8           "self": {
9             "href": "http://localhost:8080/users/admin"
10          }
11        },
12        "user": {
13          "href": "http://localhost:8080/users/admin"
14        }
15      },
16      {
17        "username": "user",
18        "email": "user@user",
19        "_links": {
20          "self": {
21            "href": "http://localhost:8080/users/user"
22          }
23        },
24        "user": {
25          "href": "http://localhost:8080/users/user"
26        }
27      },
28      {
29        "username": "artur",
30        "email": "artur@artur",
31        "_links": {
32          "self": {
33            "href": "http://localhost:8080/users/artur"
34          }
35        },
36        "user": {
37          "href": "http://localhost:8080/users/artur"
38        }
39      }
40    ]
41  },
42  "_links": {
43    "self": {
44      "href": "http://localhost:8080/users"
45    }
46  }
47 }
```

POST: users/

The screenshot shows the Postman application interface. On the left, the 'History' tab is active, displaying a list of recent requests. The main workspace shows a POST request to 'localhost:8080/users'. The request body is a JSON object with the following fields: 'username' (test), 'password' (test123456), and 'email' (test@test.com). The response is displayed in the 'Body' tab, showing a 201 Created status with a JSON response that includes a self-link and a user object with a href to the created user.

Postman

File Edit View Help

New Import Runner

My Workspace Invite

No Environment

Filter

History Collections

Clear all Save Responses

Today

- POST localhost:8080/users
- POST localhost:8080/users
- POST localhost:8080/users
- POST localhost:8080/users
- GET localhost:8080/users
- DEL localhost:8080/reservations/9
- PUT localhost:8080/reservations/9
- GET localhost:8080/reservations/
- POST localhost:8080/reservations/
- GET localhost:8080/reservations/
- POST localhost:8080/reservations/
- POST localhost:8080/reservations/
- PUT localhost:8080/reservations/2
- PUT localhost:8080/reservations/2
- GET localhost:8080/reservations/
- PUT localhost:8080/borrows/1
- PUT localhost:8080/borrows/1
- PUT localhost:8080/borrows/1
- PUT localhost:8080/borrows/1
- PUT localhost:8080/borrows/1
- PUT localhost:8080/borrows/1
- PUT localhost:8080/borrows/1
- PUT localhost:8080/borrows/1
- PUT localhost:8080/borrows/1

localhost:8080/users

POST localhost:8080/users

Send Save

Params Authorization Headers (2) Body Pre-request Script Tests Cookies Code Comments (0)

none form-data x-www-form-urlencoded raw binary JSON (application/json) Beautify

```
1 {
2   "username": "test",
3   "password": "test123456",
4   "email": "test@test.com"
5 }
```

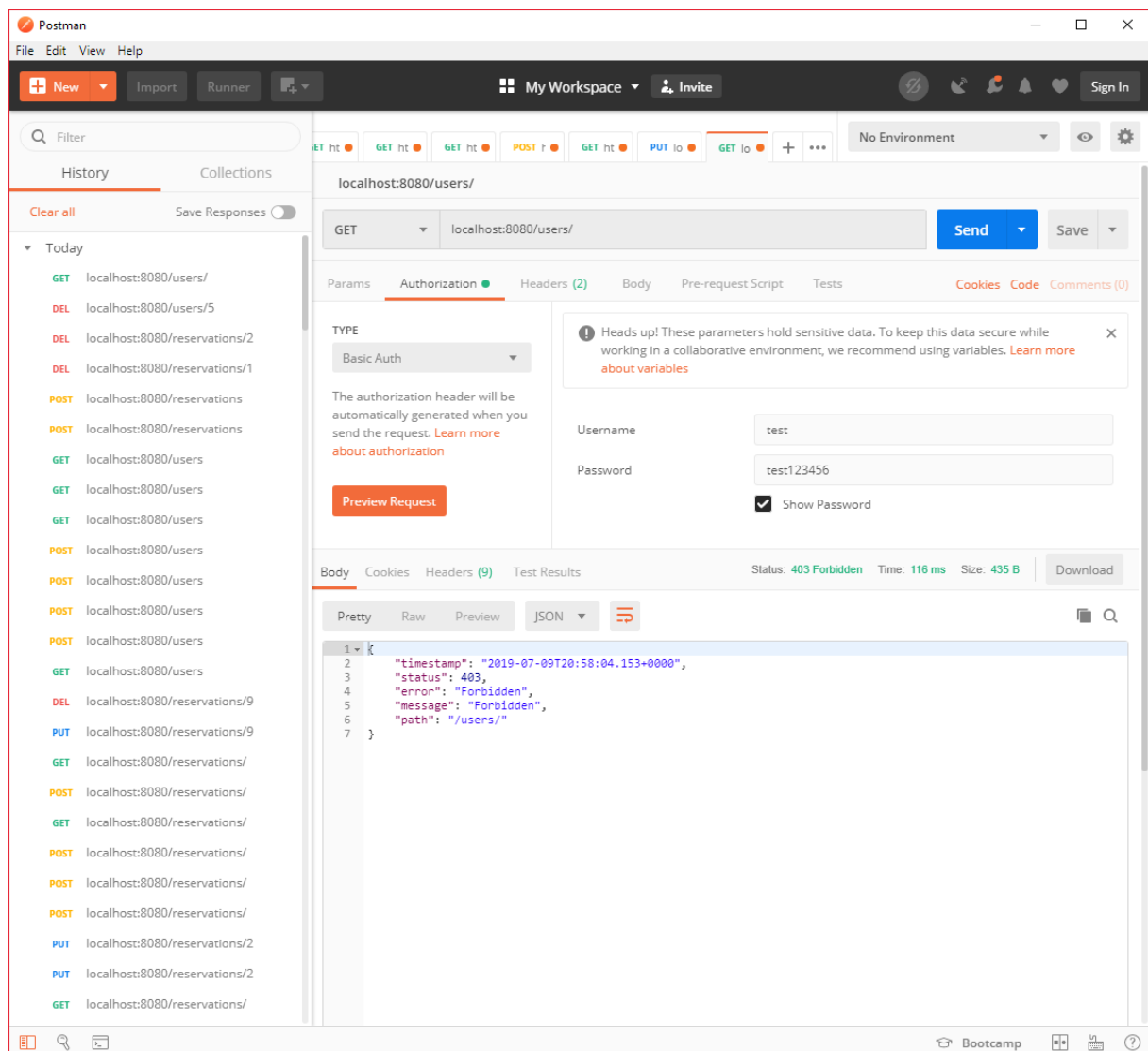
Body Cookies Headers (10) Test Results Status: 201 Created Time: 234 ms Size: 500 B Download

Pretty Raw Preview JSON

```
1 {
2   "username": "test",
3   "email": "test@test.com",
4   "_links": {
5     "self": {
6       "href": "http://localhost:8080/users/test"
7     },
8     "user": {
9       "href": "http://localhost:8080/users/test"
10    }
11  }
12 }
```

Bootcamp

Uprawnienia USER:





## Uprawnienia ADMIN:

The screenshot shows the Postman application interface. The top bar includes the Postman logo, menu options (File, Edit, View, Help), and workspace controls (My Workspace, Invite). The left sidebar displays a history of requests, with the current selection being a GET request to localhost:8080/users/. The main panel shows the details of the selected request, including the URL (localhost:8080/users/), the method (GET), and the authorization settings (Basic Auth). The authorization fields are filled with 'admin' for both username and password, and the 'Show Password' checkbox is checked. The response body is displayed in the bottom panel, showing a JSON array of users. The status of the request is 200 OK, with a response time of 118 ms and a size of 1 KB.

Postman

File Edit View Help

New Import Runner

My Workspace Invite

No Environment

Filter

History Collections

Clear all Save Responses

Today

- GET localhost:8080/users/
- GET localhost:8080/users/
- DEL localhost:8080/users/5
- DEL localhost:8080/reservations/2
- DEL localhost:8080/reservations/1
- POST localhost:8080/reservations
- POST localhost:8080/reservations
- GET localhost:8080/users
- GET localhost:8080/users
- GET localhost:8080/users
- POST localhost:8080/users
- POST localhost:8080/users
- POST localhost:8080/users
- POST localhost:8080/users
- POST localhost:8080/users
- POST localhost:8080/users
- POST localhost:8080/reservations/9
- PUT localhost:8080/reservations/9
- GET localhost:8080/reservations/
- POST localhost:8080/reservations/
- GET localhost:8080/reservations/
- POST localhost:8080/reservations/
- POST localhost:8080/reservations/
- POST localhost:8080/reservations/
- PUT localhost:8080/reservations/2
- PUT localhost:8080/reservations/2

localhost:8080/users/

GET localhost:8080/users/ Send Save

Params Authorization Headers (2) Body Pre-request Script Tests Cookies Code Comments (0)

TYPE

Basic Auth

The authorization header will be automatically generated when you send the request. [Learn more about authorization](#)

Preview Request

Heads up! These parameters hold sensitive data. To keep this data secure while working in a collaborative environment, we recommend using variables. [Learn more about variables](#)

Username admin

Password admin

☒ Show Password

Body Cookies Headers (9) Test Results Status: 200 OK Time: 118 ms Size: 1 KB Download

Pretty Raw Preview JSON

```
1 {
2   "_embedded": {
3     "users": [
4       {
5         "username": "admin",
6         "email": "admin@admin",
7         "_links": {
8           "self": {
9             "href": "http://localhost:8080/users/admin"
10          },
11          "user": {
12            "href": "http://localhost:8080/users/admin"
13          }
14        }
15      },
16      {
17        "username": "user",
18        "email": "user@user",
19        "_links": {
20          "self": {
21            "href": "http://localhost:8080/users/user"
22          },
23          "user": {
24            "href": "http://localhost:8080/users/user"
25          }
26        }
27      }
28    ]
29  }
30 }
```

Bootcamp