**Z czego się sklada link( i dlaczego?)**

Przykładowy link:

<http://localhost:3000/api/v1/auth/>

<http://localhost> - <- na razie lokalnie , później możliwe ze konfigurowalny ip, czyli jak się dostać do naszego REST – service

3000 – port aplikacji, już dziś konfigurowany w pliku application.properties : server.port=3000

/api/v1 – wersja aplikacji , na razie robimy v1 wiec można by rzec ze to stala wartosc

Auth – metoda która obsluguje jedna z zalozen CRUD-a

Linki które chcemy żeby zadzialaly :D

<http://localhost:3000/api/v1/Backlogs/>

- metoda skonfigurowana pod Get-a, daj mi wszystkie inforacje na temat wszystkich backlogi

- metoda skonfigurowana pod POST – w body musi znajdować się json o strukturze obiektu {backlog} – dodawanie nowego backlog-a

[http://localhost:3000/api/v1/Backlogs/{Backlog\_id}/](http://localhost:3000/api/v1/Backlogs/%7bBacklog_id%7d/)

- metoda skonfigurowana pod Get, daj mi wszystkie informacje na temat o {Backlog\_ID}

- metoda skonfigurowana pod DELETE – usun backlog o {Backlog\_ID}

- metoda skonfigurowana pod PUT – w update backlogu o {Backlog\_ID} w body musi znajdować się json o strukturze obiektu {backlog}

[http://localhost:3000/api/v1/Backlogs/{Backlog\_id}/UserStories/](http://localhost:3000/api/v1/Backlogs/%7bBacklog_id%7d/UserStories/)

- metoda skonfigurowana pod Get-a, daj mi wszystkie user stories danego backloga

- metoda skonfigurowana pod POST – w body musi znajdować się json o strukturze obiektu {UserStory} – dodawanie nowego UserStory

[http://localhost:3000/api/v1/Backlogs/{Backlog\_id}/UserStories/{UserStoriesId}/](http://localhost:3000/api/v1/Backlogs/%7bBacklog_id%7d/UserStories/%7bUserStoriesId%7d/)

- metoda skonfigurowana pod Get, daj mi wszystkie informacje na temat o { UserStoriesId}

- metoda skonfigurowana pod DELETE – usun user story o { UserStoriesId }

- metoda skonfigurowana pod PUT – w update user story o { UserStoriesId } w body musi znajdować się json o strukturze obiektu { UserStory }

[http://localhost:3000/api/v1/Backlogs/{Backlog\_id}/UserStories/{UserStoriesId}/Tasks/](http://localhost:3000/api/v1/Backlogs/%7bBacklog_id%7d/UserStories/%7bUserStoriesId%7d/Tasks/)

- metoda skonfigurowana pod Get-a, daj mi wszystkie taski danego userstory

- metoda skonfigurowana pod POST – w body musi znajdować się json o strukturze obiektu {Task} – dodawanie nowego Tasku

<http://localhost:3000/api/v1/Backlogs/{Backlog_id}/UserStories/{UserStoriesId}/Tasks/>{TaskId}

- metoda skonfigurowana pod Get, daj mi wszystkie informacje na temat o { TaskId }

- metoda skonfigurowana pod DELETE – usun task o { TaskId }

- metoda skonfigurowana pod PUT – w update task o { TaskId } w body musi znajdować się json o strukturze obiektu { Task }