

Project management tool - Dokumentacja

Spis treści

Opis projektu

Wstęp

Wykonanie projektu

Opis środowiska deweloperskiego

Workflow projektu

Autorzy

Resources

Account-resource

isAuthenticated

activateAccount

registerAccount

invalidateSession

getCurrentSessions

changePassword

finishPasswordReset

getAccount

saveAccount

requestPasswordReset

Audit-resource

getAll

get

Board-resource

updateBoard

createBoard

getBoard

Logs-resource

changeLevel

getList

Project-resource

getProject

deleteProject

updateProject

getAllProjectsForCurrentUser

createProject

addUserToProject

deleteUserFromProject

Task-list-resource

updateTaskList

createTaskList

getTaskList

Task-resource

updateTask

createTask

getTask

User-resource

getUser

deleteUser

updateUser

getAllUsers

createUser

Definitions

AuditEvent

BoardDTO

BoardShortDTO

KeyAndPasswordDTO

LoggerDTO

ManagedUserDTO

PersistentToken

ProjectDTO

TaskDTO

TaskListDTO

UserDTO

Opis projektu

Wstęp

Tematem projektu było stworzenie systemu służącego do zarządzania projektami informatycznymi. System ten został nazwany Project Management Tool. W założeniu, system miał pozwalać na tworzenie grup developerskich, skupionych wokół prywatnych bądź publicznych projektów. W obrębie tych projektów członkowie grup mogli definiować swoje tablice (np. związane z konkretnym sprintem, częścią projektu), w obrębie każdej z tych tablic możliwe było definiowanie list zadań (np. „w trakcie”, „do zrobienia”, „zakończone”), a w obrębie każdej z tych list miało być możliwe definiowanie zadań. Każde z zadań mogłoby zostać przydzielone do konkretnej osoby, a każda z pracujących nad nim osób mogłaby zalogować swój czas pracy nad tym zadaniem.

Wykonanie projektu

Projekt podzielony jest na dwie części:

- Back-end – cała logika aplikacji, REST Api, serwisy
- Front-end – strona projektu, wyświetlana za pomocą przeglądarki

Back-end został w całości napisany przy pomocy języka Java. Frameworkiem zapewniającym biblioteki do większości potrzebnych funkcjonalności jest Spring. Skorzystaliśmy z narzędzia Springboot, a do testów wykorzystane zostały frameworki JUnit oraz Hamcrest

Front-end został napisany przy użyciu HTML5, CSS oraz języka Javascript, Javascriptowym frameworkiem wykorzystanym przy tworzeniu front-endu jest AngularJS. Front-end aplikacji jest aplikacją typu SPA (Single Page Application).

Opis środowiska deweloperskiego

- Projekt został stworzony z użyciem generatora projektów Jhipster. Wytyczne podanie przy tworzeniu projektu:
 - Type: monolithic application
 - Default java package name: pw.elka.pik.mkdev1
 - Authentication type: classical HTTP session authentication
 - Dependency manager: Maven
 - Production database type: MySQL
 - Developmnet database type: MySQLn
 - Hibernate 2nd level cache
 - No search engine
 - No HTTP clustered sessions
 - No Websockets
- Używane środowiska programistyczne
 - Eclipse
 - IntelliJ IDEA

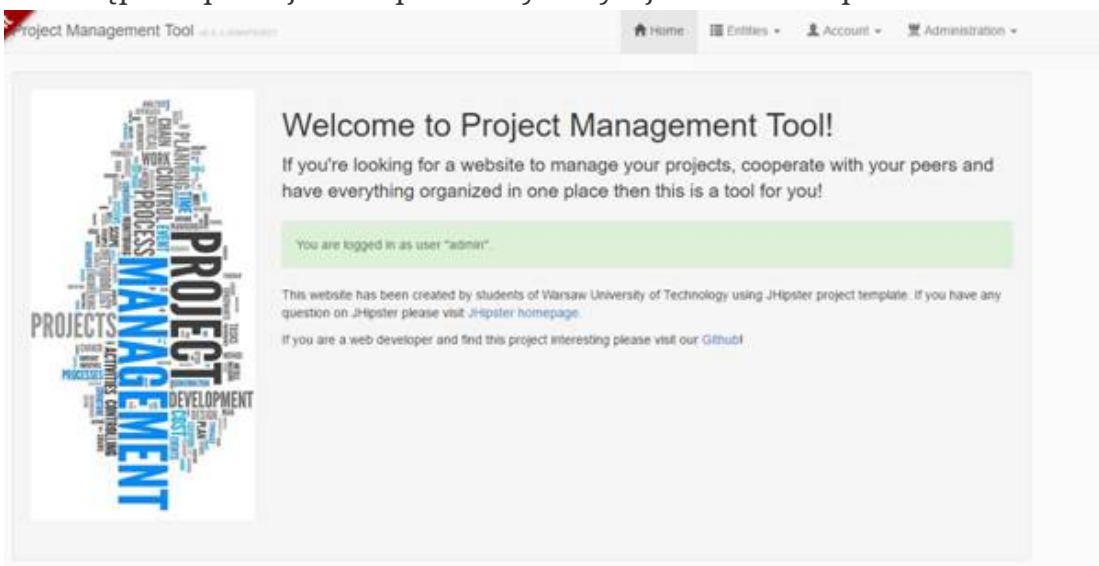
- Repozytorium kodu, a zarazem narzędziem przez które prowadzone było code review jest Github.
 - Adres repozytorium: <https://github.com/mmudel/Pw.Elka.Pik.Mkdev1>
 - Dostęp do repozytorium jest publiczny
- Serwerem, na którym stawiane były narzędzia służące do wspomagania pracy nad projektem, a także wystawiony był sam projekt był serwer na wydziale Elektroniki i Technik Informatycznych
 - Adres serwera: pik07.ii.pw.edu.pl
 - Jedyny dostęp do serwera był możliwy poprzez inny serwer na tym samym wydziale: galera.ii.pw.edu.pl
- W celu dostępu do narzędzi skonfigurowaliśmy tunelowanie. Na systemach Windows użyliśmy w tym celu programu MobaXterm



- Do Ciągłej Integracji skorzystaliśmy z narzędzia Jenkins.
 - Dostęp do narzędzia na wspomnianym wyżej serwerze na porcie 9090
 - Jenkins został skonfigurowany w ten sposób, aby utworzył oddzielny job na każdy z branch'y projektu.
 - Nie został skonfigurowany automatyczny build projektu przy każdym wypchnięciu zmian do repozytorium ze względu na utrudniony dostęp do serwera, na którym wystawiony jest Jenkins
 - Zamiast automatycznego budowania gałęzi po wypchnięciu zmian, skonfigurowany został mechanizm, który powoduje, że co 15 min Jenkins sam odpytuje repozytorium czy były jakieś zmiany i w razie jeśli były, buduje odpowiednią gałąź
 - Pomyślne wykonanie zbudowania gałęzi master powoduje uruchomienie innego joba, którego zadaniem jest automatyczny deploy aplikacji na serwer aplikacji Apache Tomcat

S	W	Nazwa	Ostatni sukces	Ostatni błąd	Czas trwania
		feature/add&delete-user-from-project	7 dni 15 godz - #4	—	53 sek
		feature/add-new-project	1 mies 6 dni - #2	—	1 min 47 sek
		feature/delete-project	2 dni 17 godz - #1	—	1 min 10 sek
		feature/entities-edit&create	8 dni 1 godz - #5	—	51 sek
		feature/modify-project-details	9 dni 23 godz - #4	—	49 sek
		feature/projects-all	2 dni 19 godz - #1	—	1 min 10 sek
		feature/show-board-details	10 dni - #5	—	48 sek
		feature/show-boards	17 dni - #3	—	1 min 35 sek
		feature/show-project-details	10 dni - #3	—	48 sek
		fix/deploy-on-server	2 dni 23 godz - #5	4 dni 22 godz - #1	1 min 13 sek
		fix/projects-service1	16 dni - #2	—	1 min 54 sek
		fix/user-see-only-his-projects	8 dni 16 godz - #4	—	1 min 3 sek
		master	8 min 16 sek - #93	12 min - #91	1 min 16 sek

- Jako bazy danych użyliśmy serwera bazy danych MySQL
 - Dostęp do bazy danych na wspomnianym wyżej serwerze na porcie 3306
 - Lokalnie w celu połączenia z bazą korzystaliśmy z narzędzia MySQL Workbench
- Aplikacja jest uruchamiana za pomocą narzędzia Apache Tomcat.
 - Dostęp do aplikacji na wspomnianym wyżej serwerze na porcie 8080



Workflow projektu

W projekcie został wspólnie zdefiniowany następujący workflow, dotyczący developowania projektu:

1. Wybieramy jedno z user story w trello i przeciągamy je do kolumny In progress.
2. Zmiany wprowadzamy w branchu związanym z daną funkcjonalnością, jeśli taki branch nie istnieje to tworzymy nowy – feature/nazwa_funkcjonalności.
3. Po wprowadzeniu całej funkcjonalności sprawdzamy czy branch buduje się
 - W używanym IDE
 - Po wpisaniu w konsoli znajdując się w głównym katalogu projektu `mvn clean install`
 - W systemie Jenkins (odpowiedni job znajduje się w folderze mkdev1).

Jeśli branch buduje się bez błędów, możemy zmergować go z masterem, a następnie uruchomić job budujący gałąź master na Jenkinsie.

4. W Trello przierzucamy odpowiednie zadanie do listy Done. W komentarzu przy zadaniu podajemy hashe commitów oraz nazwę brancha który dotyczy tego zadania.

Autorzy

- Michał Mudel
- Joanna Raczyńska
- Wojciech Zieliński

Projekt został stworzony w ramach przedmiotu Programowanie Aplikacji Internetowych i Korporacyjnych prowadzonego na wydziale Elektroniki i Technik Informatycznych Politechniki Warszawskiej.

Resources

Account-resource

Account Resource

isAuthenticated

GET /api/authenticate

Responses

HTTP Code	Description	Schema
200	OK	string
401	Unauthorized	No Content
403	Forbidden	No Content
404	Not Found	No Content

Consumes

- application/json

Produces

- application/json

activateAccount

GET /api/activate

Parameters

Type	Name	Description	Required	Schema	Default
QueryParameter	key	key	true	string	

Responses

HTTP Code	Description	Schema
200	OK	string
401	Unauthorized	No Content
403	Forbidden	No Content

404	Not Found	No Content
-----	-----------	------------

Consumes

- application/json

Produces

- application/json

registerAccount

POST /api/register

Parameters

Type	Name	Description	Required	Schema	Default
BodyParameter	userDTO	userDTO	true	UserDTO	

Responses

HTTP Code	Description	Schema
200	OK	object
201	Created	No Content
401	Unauthorized	No Content
403	Forbidden	No Content
404	Not Found	No Content

Consumes

- application/json

Produces

- application/json
- text/plain

invalidateSession

DELETE /api/account/sessions/{series}

Parameters

Type	Name	Description	Required	Schema	Default

PathParamter	series	series	true	string	
--------------	--------	--------	------	--------	--

Responses

HTTP Code	Description	Schema
200	OK	No Content
204	No Content	No Content
401	Unauthorized	No Content
403	Forbidden	No Content

Consumes

- application/json

Produces

- /

getCurrentSessions

GET /api/account/sessions

Responses

HTTP Code	Description	Schema
200	OK	PersistentToken array
401	Unauthorized	No Content
403	Forbidden	No Content
404	Not Found	No Content

Consumes

- application/json

Produces

- application/json

changePassword

POST /api/account/change_password

Parameters

--	--	--	--	--	--

Type	Name	Description	Required	Schema	Default
BodyParameter	password	password	true	string	

Responses

HTTP Code	Description	Schema
200	OK	object
201	Created	No Content
401	Unauthorized	No Content
403	Forbidden	No Content
404	Not Found	No Content

Consumes

- application/json

Produces

- text/plain

finishPasswordReset

POST /api/account/reset_password/finish

Parameters

Type	Name	Description	Required	Schema	Default
BodyParameter	keyAndPassword	keyAndPassword	true	KeyAndPasswordDTO	

Responses

HTTP Code	Description	Schema
200	OK	string
201	Created	No Content
401	Unauthorized	No Content
403	Forbidden	No Content
404	Not Found	No Content

Consumes

- application/json

Produces

- text/plain

getAccount

GET /api/account

Responses

HTTP Code	Description	Schema
200	OK	UserDTO
401	Unauthorized	No Content
403	Forbidden	No Content
404	Not Found	No Content

Consumes

- application/json

Produces

- application/json

saveAccount

POST /api/account

Parameters

Type	Name	Description	Required	Schema	Default
BodyParameter	userDTO	userDTO	true	UserDTO	

Responses

HTTP Code	Description	Schema
200	OK	string
201	Created	No Content
401	Unauthorized	No Content

403	Forbidden	No Content
404	Not Found	No Content

Consumes

- application/json

Produces

- application/json

requestPasswordReset

POST /api/account/reset_password/init

Parameters

Type	Name	Description	Required	Schema	Default
BodyParameter	mail	mail	true	string	

Responses

HTTP Code	Description	Schema
200	OK	object
201	Created	No Content
401	Unauthorized	No Content
403	Forbidden	No Content
404	Not Found	No Content

Consumes

- application/json

Produces

- text/plain

Audit-resource

Audit Resource

getAll

GET /api/audits

Responses

HTTP Code	Description	Schema
200	OK	AuditEvent array
401	Unauthorized	No Content
403	Forbidden	No Content
404	Not Found	No Content

Consumes

- application/json

Produces

- application/json

get

GET /api/audits/{id}

Parameters

Type	Name	Description	Required	Schema	Default
PathParameter	id	id	true	integer (int64)	

Responses

HTTP Code	Description	Schema
200	OK	AuditEvent
401	Unauthorized	No Content
403	Forbidden	No Content
404	Not Found	No Content

Consumes

- application/json

Produces

- application/json

Board-resource

Board Resource

updateBoard

PUT /api/boards

Parameters

Type	Name	Description	Required	Schema	Default
BodyParameter	boardDTO	boardDTO	true	BoardDTO	

Responses

HTTP Code	Description	Schema
200	OK	object
201	Created	No Content
401	Unauthorized	No Content
403	Forbidden	No Content
404	Not Found	No Content

Consumes

- application/json

Produces

- application/json

createBoard

POST /api/boards

Parameters

Type	Name	Description	Required	Schema	Default
BodyParameter	boardDTO	boardDTO	true	BoardDTO	

Responses

HTTP Code	Description	Schema
200	OK	object

201	Created	No Content
401	Unauthorized	No Content
403	Forbidden	No Content
404	Not Found	No Content

Consumes

- application/json

Produces

- application/json

getBoard

GET /api/boards/{boardId}

Parameters

Type	Name	Description	Required	Schema	Default
PathParameter	boardId	boardId	true	integer (int64)	

Responses

HTTP Code	Description	Schema
200	OK	BoardDTO
401	Unauthorized	No Content
403	Forbidden	No Content
404	Not Found	No Content

Consumes

- application/json

Produces

- application/json

Logs-resource

Logs Resource

changeLevel

PUT /api/logs

Parameters

Type	Name	Description	Required	Schema	Default
BodyParameter	jsonLogger	jsonLogger	true	LoggerDTO	

Responses

HTTP Code	Description	Schema
201	Created	No Content
204	No Content	No Content
401	Unauthorized	No Content
403	Forbidden	No Content
404	Not Found	No Content

Consumes

- application/json

Produces

- /

getList

GET /api/logs

Responses

HTTP Code	Description	Schema
200	OK	LoggerDTO array
401	Unauthorized	No Content
403	Forbidden	No Content
404	Not Found	No Content

Consumes

- application/json

Produces

- application/json

Project-resource

Project Resource

getProject

GET /api/projects/{id}

Parameters

Type	Name	Description	Required	Schema	Default
PathParameter	id	id	true	integer (int64)	

Responses

HTTP Code	Description	Schema
200	OK	ProjectDTO
401	Unauthorized	No Content
403	Forbidden	No Content
404	Not Found	No Content

Consumes

- application/json

Produces

- application/json

deleteProject

DELETE /api/projects/{id}

Parameters

Type	Name	Description	Required	Schema	Default
PathParameter	id	id	true	integer (int64)	

Responses

HTTP Code	Description	Schema
-----------	-------------	--------

200	OK	No Content
204	No Content	No Content
401	Unauthorized	No Content
403	Forbidden	No Content

Consumes

- application/json

Produces

- application/json

updateProject

PUT /api/projects

Parameters

Type	Name	Description	Required	Schema	Default
BodyParameter	projectDTO	projectDTO	true	ProjectDTO	

Responses

HTTP Code	Description	Schema
200	OK	ProjectDTO
201	Created	No Content
401	Unauthorized	No Content
403	Forbidden	No Content
404	Not Found	No Content

Consumes

- application/json

Produces

- application/json

getAllProjectsForCurrentUser

GET /api/projects

Responses

HTTP Code	Description	Schema
200	OK	ProjectDTO array
401	Unauthorized	No Content
403	Forbidden	No Content
404	Not Found	No Content

Consumes

- application/json

Produces

- application/json

createProject

POST /api/projects

Parameters

Type	Name	Description	Required	Schema	Default
BodyParameter	projectDTO	projectDTO	true	ProjectDTO	

Responses

HTTP Code	Description	Schema
200	OK	object
201	Created	No Content
401	Unauthorized	No Content
403	Forbidden	No Content
404	Not Found	No Content

Consumes

- application/json

Produces

- application/json

addUserToProject

POST /api/projects/{id}/{login}

Parameters

Type	Name	Description	Required	Schema	Default
PathParameter	id	id	true	integer (int64)	
PathParameter	login	login	true	string	

Responses

HTTP Code	Description	Schema
200	OK	No Content
201	Created	No Content
401	Unauthorized	No Content
403	Forbidden	No Content
404	Not Found	No Content

Consumes

- application/json

Produces

- /

deleteUserFromProject

DELETE /api/projects/{id}/{login}

Parameters

Type	Name	Description	Required	Schema	Default
PathParameter	id	id	true	integer (int64)	
PathParameter	login	login	true	string	

Responses

HTTP Code	Description	Schema

200	OK	No Content
204	No Content	No Content
401	Unauthorized	No Content
403	Forbidden	No Content

Consumes

- application/json

Produces

- /

Task-list-resource

Task List Resource

updateTaskList

PUT /api/tasklists

Parameters

Type	Name	Description	Required	Schema	Default
BodyParameter	taskListDTO	taskListDTO	true	TaskListDTO	

Responses

HTTP Code	Description	Schema
200	OK	object
201	Created	No Content
401	Unauthorized	No Content
403	Forbidden	No Content
404	Not Found	No Content

Consumes

- application/json

Produces

- application/json

createTaskList

POST /api/tasklists

Parameters

Type	Name	Description	Required	Schema	Default
BodyParameter	taskListDTO	taskListDTO	true	TaskListDTO	

Responses

HTTP Code	Description	Schema
200	OK	object
201	Created	No Content
401	Unauthorized	No Content
403	Forbidden	No Content
404	Not Found	No Content

Consumes

- application/json

Produces

- application/json

getTaskList

GET /api/tasklists/{taskListId}

Parameters

Type	Name	Description	Required	Schema	Default
PathParameter	taskListId	taskListId	true	integer (int64)	

Responses

HTTP Code	Description	Schema
200	OK	object
401	Unauthorized	No Content

403	Forbidden	No Content
404	Not Found	No Content

Consumes

- application/json

Produces

- application/json

Task-resource

Task Resource

updateTask

PUT /api/tasks

Parameters

Type	Name	Description	Required	Schema	Default
BodyParameter	taskDTO	taskDTO	true	TaskDTO	

Responses

HTTP Code	Description	Schema
200	OK	object
201	Created	No Content
401	Unauthorized	No Content
403	Forbidden	No Content
404	Not Found	No Content

Consumes

- application/json

Produces

- application/json

createTask

POST /api/tasks

Parameters

Type	Name	Description	Required	Schema	Default
BodyParameter	taskDTO	taskDTO	true	TaskDTO	

Responses

HTTP Code	Description	Schema
200	OK	object
201	Created	No Content
401	Unauthorized	No Content
403	Forbidden	No Content
404	Not Found	No Content

Consumes

- application/json

Produces

- application/json

getTask

GET /api/tasks/{taskId}

Parameters

Type	Name	Description	Required	Schema	Default
PathParameter	taskId	taskId	true	integer (int64)	

Responses

HTTP Code	Description	Schema
200	OK	object
401	Unauthorized	No Content
403	Forbidden	No Content
404	Not Found	No Content

Consumes

- application/json

Produces

- application/json

User-resource

User Resource

getUser

GET /api/users/{login}

Parameters

Type	Name	Description	Required	Schema	Default
PathParameter	login	login	true	string	

Responses

HTTP Code	Description	Schema
200	OK	ManagedUserDTO
401	Unauthorized	No Content
403	Forbidden	No Content
404	Not Found	No Content

Consumes

- application/json

Produces

- application/json

deleteUser

DELETE /api/users/{login}

Parameters

Type	Name	Description	Required	Schema	Default
PathParameter	login	login	true	string	

Responses

HTTP Code	Description	Schema
200	OK	No Content
204	No Content	No Content
401	Unauthorized	No Content
403	Forbidden	No Content

Consumes

- application/json

Produces

- application/json

updateUser

PUT /api/users

Parameters

Type	Name	Description	Required	Schema	Default
BodyParameter	managedUserDTO	managedUserDTO	true	ManagedUserDTO	

Responses

HTTP Code	Description	Schema
200	OK	ManagedUserDTO
201	Created	No Content
401	Unauthorized	No Content
403	Forbidden	No Content
404	Not Found	No Content

Consumes

- application/json

Produces

- application/json

getAllUsers

GET /api/users

Responses

HTTP Code	Description	Schema
200	OK	ManagedUserDTO array
401	Unauthorized	No Content
403	Forbidden	No Content
404	Not Found	No Content

Consumes

- application/json

Produces

- application/json

Example CURL request

\$ curl 'http://localhost:8080/api/users' -i -H 'Accept: application/json'

BASH

Example HTTP request

GET /api/users HTTP/1.1
Accept: application/json
Host: localhost

HTTP

Example HTTP response

HTTP/1.1 200 OK
X-Total-Count: 6
Link: </api/users?page=0&size=20>; rel="last",</api/users?page=0&size=20>; rel="first"
Content-Type: application/json
Content-Length: 2084

```
[ {
  "login" : "system",
  "password" : null,
  "firstName" : "System",
  "lastName" : "System",
  "email" : "system@localhost",
  "activated" : true,
  "langKey" : "en",
  "authorities" : [ "ROLE_USER", "ROLE_ADMIN" ],
  "id" : 1,
  "createdDate" : "2016-05-28T14:03:57.000Z",
  "lastModifiedBy" : null,
  "lastModifiedDate" : null
}, {
  "login" : "anonymousUser",
  "password" : null,
  "firstName" : "Anonymous",
  "lastName" : "User",
  "email" : "anonymous@localhost",
  "activated" : true,
  "langKey" : "en",
  "authorities" : [ ],
  "id" : 2,
  "createdDate" : "2016-05-28T14:03:57.000Z",
  "lastModifiedBy" : null,
  "lastModifiedDate" : null
}, {
  "login" : "admin",
  "password" : null,
  "firstName" : "Administrator",
  "lastName" : "Administrator",
  "email" : "admin@localhost",
  "activated" : true,
  "langKey" : "en",
  "authorities" : [ "ROLE_USER", "ROLE_ADMIN" ],
  "id" : 3,
  "createdDate" : "2016-05-28T14:03:57.000Z",
  "lastModifiedBy" : null,
  "lastModifiedDate" : null
}, {
  "login" : "user",
  "password" : null,
  "firstName" : "User",
  "lastName" : "User",
  "email" : "user@localhost",
  "activated" : true,
  "langKey" : "en",
  "authorities" : [ "ROLE_USER" ],
  "id" : 4,
  "createdDate" : "2016-05-28T14:03:57.000Z",
  "lastModifiedBy" : null,
  "lastModifiedDate" : null
}, {
  "login" : "testuser",
  "password" : null,
```

```

    "firstName" : null,
    "lastName" : null,
    "email" : "testmail@testmail.com",
    "activated" : true,
    "langKey" : "en",
    "authorities" : [ "ROLE_USER" ],
    "id" : 5,
    "createdDate" : "2016-05-28T23:36:56.000Z",
    "lastModifiedBy" : "admin",
    "lastModifiedDate" : "2016-05-28T23:37:14.000Z"
  }, {
    "login" : "newuser5",
    "password" : null,
    "firstName" : null,
    "lastName" : null,
    "email" : "newuser5@newuser5.pl",
    "activated" : true,
    "langKey" : "en",
    "authorities" : [ "ROLE_USER" ],
    "id" : 6,
    "createdDate" : "2016-06-03T10:18:49.000Z",
    "lastModifiedBy" : "admin",
    "lastModifiedDate" : "2016-06-03T10:19:01.000Z"
  } ]

```

createUser

POST /api/users

Parameters

Type	Name	Description	Required	Schema	Default
BodyParameter	managedUserDTO	managedUserDTO	true	ManagedUserDTO	

Responses

HTTP Code	Description	Schema
200	OK	object
201	Created	No Content
401	Unauthorized	No Content
403	Forbidden	No Content
404	Not Found	No Content

Consumes

- application/json

Produces

- application/json

Definitions

AuditEvent

Name	Description	Required	Schema	Default
data		false	object	
principal		false	string	
timestamp		false	string (date-time)	
type		false	string	

BoardDTO

Name	Description	Required	Schema	Default
id		false	integer (int64)	
name		false	string	
projectId		false	integer (int64)	
taskLists		false	TaskListDTO array	

BoardShortDTO

Name	Description	Required	Schema	Default
id		false	integer (int64)	
name		false	string	

KeyAndPasswordDTO

Name	Description	Required	Schema	Default
key		false	string	
newPassword		false	string	

LoggerDTO

Name	Description	Required	Schema	Default
level		false	string	

name		false	string	
------	--	-------	--------	--

ManagedUserDTO

Name	Description	Required	Schema	Default
activated		false	boolean	
authorities		false	string array	
createdDate		false	string (date-time)	
email		false	string	
firstName		false	string	
id		false	integer (int64)	
langKey		false	string	
lastModifiedBy		false	string	
lastModifiedDate		false	string (date-time)	
lastName		false	string	
login		false	string	
password		false	string	

PersistentToken

Name	Description	Required	Schema	Default
formattedTokenDate		false	string	
ipAddress		false	string	
series		false	string	
userAgent		false	string	

ProjectDTO

Name	Description	Required	Schema	Default
boards		false		

			BoardShortDTO array	
description		false	string	
id		false	integer (int64)	
membersCount		false	integer (int64)	
name		false	string	
users		false	string array	

TaskDTO

Name	Description	Required	Schema	Default
id		false	integer (int64)	
name		false	string	
taskListId		false	integer (int64)	

TaskListDTO

Name	Description	Required	Schema	Default
boardId		false	integer (int64)	
id		false	integer (int64)	
name		false	string	
tasks		false	TaskDTO array	

UserDTO

Name	Description	Required	Schema	Default
activated		false	boolean	
authorities		false	string array	
email		false	string	
firstName		false	string	
langKey		false	string	

lastName		false	string	
login		false	string	
password		false	string	

Last updated 2016-06-03 22:54:44 CEST