REQUIREMENTS

# Persistence

## MySQL

1. MySQLUserPersistence.java (JOCHEN)

|  |  |
| --- | --- |
| Methodenname | Public List<ResearchProject> getProjectsByUser(int id, boolean lazy) |
| Input | Int id, boolean lazy |
| Return | projects |
| Beschreibung | Gibt die Projekte, an welchen der User interessiert ist und von ResearchProject abstammen (papers, offers, finalpapers), - abhängig von der übergebenen id - zurück |

|  |  |
| --- | --- |
| Methodenname | Public List <Role> getRolesByUser(int id, boolean lazy) |
| Input | Int id, boolean lazy |
| Return | roles |
| Beschreibung | Gibt die Rollen, die von Roles abstammen (teacher, student, supervisor, secretary, admin) - abhängig von der übergebenen id - zurück |

|  |  |
| --- | --- |
| Methodenname | Public boolean hasUserWithLogin (String Login) |
| Input | String login |
| Beschreibung | Prüft, ob ein bestimmter User mit Login in der Datenbank existiert. Außerdem wird auf inaktive (gelöschte) User geprüft und liefert "true" zurück, wenn ein gelöschter User mit dem gelieferten Login in der Datenbank existiert. |

|  |  |
| --- | --- |
| Methodenname | Public List <User> getAllUsers() |
| Return | users |
| Beschreibung | Gibt alle User aus der Datenbank zurück, deren Status 0 (nicht gelöscht) ist, ansonsten wird eine Exception geworfen. |

|  |  |
| --- | --- |
| Methodenname | Public User getUserById(int id) |
| Input | Int id |
| Return | user |
| Beschreibung | Soll User mit einer bestimmten Id aus der Datenbank zurückgeben, ansonsten wird eine Exception geworfen. |

|  |  |
| --- | --- |
| Methodenname | Public User getUserByEmail(String email) |
| Input | String email |
| Return | user |
| Beschreibung | Soll User mit einer bestimmten Email aus der Datenbank zurückgegen, ansonsten wird Exception geworfen. |

|  |  |
| --- | --- |
| Methodenname | Public int addUser(User user) |
| Return | key |
| Beschreibung | Anlegen eines neuen Users mit Parametern (login, password, iterations, salt, firstname, lastname, title, email) |

|  |  |
| --- | --- |
| Methodenname | Public void editUser(User user) |
| Beschreibung | Update eines Users mit bestimmter Id der Paramater (login, firstname, lastname, title, email) |

|  |  |
| --- | --- |
| Methodenname | Public void deleteUserById(int id) |
| Input | int id |
| Beschreibung | Löscht einen User mit bestimmter Id, indem Status = 1 (gelöscht) gesetzt wird. |

2.MySQLStudentPersistence.java (LARS)

|  |  |
| --- | --- |
| Methodenname | public List<Student> getRolesByUserId (int userId) |
| Input | int userId |
| Return | Liste mit student Objekten |
| Beschreibung | Speichert alle student Objekte mit dem Input userId. (Für jede Rolle wird ein neues student Objekt genutzt) |

|  |  |
| --- | --- |
| Methodenname | public Student getRoleById(int id) |
| Input | student Id |
| Return | student Objekt |
| Beschreibung | Liefert student Objekt mit der Input Id ????? |

|  |  |
| --- | --- |
| Methodenname | public int addRole(Student role) |
| Input | student Objekt |
| Return | key als int |
| Beschreibung | Die als Student Objekt überlieferte Rolle wird in die DB geschrieben. |

|  |  |
| --- | --- |
| Methodenname | public void editRole(Student role) |
| Input | student Objekt |
| Return | - |
| Beschreibung | Ändert (Update) das student Objekt in der DB. |

|  |  |
| --- | --- |
| Methodenname | public void deleteRoleById(int id) |
| Input | student Id |
| Return | - |
| Beschreibung | Setzt das end\_date des Student Objekts mit der übergebenen Id auf gestern. |

3. MySQLTeacherPersistence.java (JOCHEN)

|  |  |
| --- | --- |
| Methodenname | Public List <Teacher> getRolesByUserId(int userId) |
| Input | int userId |
| Return | teachers |
| Beschreibung | Gibt die zugehörigen Dozenten-Rollen zu bestimmten User Id's zurück, ansonsten wird eine Exception geworfen, falls die Id's nicht gefunden wurden. |

|  |  |
| --- | --- |
| Methodenname | Public Teacher getRoleById(int id) |
| Input | int id |
| Return | teacher |
| Beschreibung | Gibt die Dozenten-Rolle zu einer bestimmten Id zurück, ansonsten wird eine Exception geworfen, falls die Id nicht gefunden wurde. |

|  |  |
| --- | --- |
| Methodenname | Public int addRole(Teacher role) |
| Return | key |
| Beschreibung | Anlegen eines neuen Dozenten mit Parametern (user\_id, start\_date, end\_date, room, phone), ansonsten wird eine Exception geworfen, falls die Id nicht gefunden wurde oder der Vorgang fehlschlug. |

|  |  |
| --- | --- |
| Methodenname | Public void editRole(Teacher role) |
| Beschreibung | Update eines Dozenten mit bestimmter Id der Parameter (start\_date, end\_date, room, phone), ansonsten wird eine Exception geworfen, falls die Id nicht gefunden wurde oder der Vorgang fehlschlug. |

|  |  |
| --- | --- |
| Methodenname | Public void deleteRoleById(int id) |
| Input | int id |
| Beschreibung | Löscht einen Dozenten mit bestimmter Id, indem der Parameter end\_date upgedatet wird, ansonsten wird eine Exception geworfen, falls die Id nicht gefunden wurde oder der Vorgang fehlschlug. |

4. MySQLResearchProjectOfferPersistence.java (JOCHEN)

|  |  |
| --- | --- |
| Methodenname | Public List <ResearchProjectOffer> getAllProjects() |
| Return | offers |
| Beschreibung | Es werden alle Angebote aus der Datenbank zurückgegeben, deren Status visible =1 (sichtbar) und deleted = 0 (nicht gelöscht) sind, ansonsten wird eine Exception geworfen, dass nicht alle Angebote zurückgegeben werden konnten. |

|  |  |
| --- | --- |
| Methodenname | Public ResearchProjectOffer getProjectById(int id) |
| Input | int id |
| Return | offer |
| Beschreibung | Es wird ein Angebot aus der Datenbank zu einer bestimmten Id zurückgegeben, deren Status deleted = 0 (nicht gelöscht) ist, ansonsten wird eine Exception geworfen, dass kein Angebot zur angegebenen Id gefunden wurde. |

|  |  |
| --- | --- |
| Methodenname | Public ResearchProjectOffer getProjectByUuid(string uuid) |
| Input | string uuid |
| Return | offer |
| Beschreibung | Es wird ein Angebot aus der Datenbank zu einer bestimmten ResearchProjectOfferUuid zurückgegeben, deren Status deleted = 0 (nicht gelöscht) ist, ansonsten wird eine Exception geworfen, dass kein Angebot zur angegebenen Id gefunden wurde. |

|  |  |
| --- | --- |
| Methodenname | Public int addProject(ResearchProjectOffer project) |
| Return | key |
| Beschreibung | Hinzufügen eines neuen Angebots in die Datenbank mit den Parametern (title, creator\_user\_id, description, description\_long, research\_project\_id, visible), ansonsten wird eine Exception geworfen, dass der Vorgang fehlschlug. |

|  |  |
| --- | --- |
| Methodenname | Public void editProject(ResearchProjectOffer project) |
| Beschreibung | Update eines Angebots mit bestimmter Id der Parameter (title, creator\_user\_id, description, description\_long, research\_project\_id, visible), ansonsten wird eine Exception geworfen, falls die Id nicht gefunden wurde oder der Vorgang fehlschlug. |

|  |  |
| --- | --- |
| Methodenname | Public void deleteProjectById (int id) |
| Input | int id |
| Beschreibung | Löscht ein Angebot mit bestimmter Id, indem der Parameter deleted = 1 (gelöscht) gesetzt wird, ansonsten wird eine Exception geworfen, falls die Id nicht gefunden wurde. |

|  |  |
| --- | --- |
| Methodenname | Public List<ResearchProjectOffer> getProjectsByUserId(int id) |
| Input | int id |
| Return | offers |
| Beschreibung | Es werden alle Angebote zu einer bestimmten User Id aus der Datenbank zurückgegeben, ansonsten wird eine Exception geworfen, dass nicht alle Angebote zu der eingegebenen User Id gefunden wurden konnten. |

|  |  |
| --- | --- |
| Methodenname | Public List<User> getInterestedUsers(int offerId) |
| Input | int offerId |
| Return | interestedUsers |
| Beschreibung | Es werden alle interessierten User aus der Datenbank, welche an einem bestimmten Angebot interessiert sind, zurückgegeben, ansonsten wird eine Exception zurückgegeben, dass keine User zu einem Angebot gefunden werden konnten. |

5.MySQLThreadPersistence.java (LARS)

|  |  |
| --- | --- |
| Methodenname | public List<Thread> getAllThreadsByResearchProjectId(int id, boolean lazy) |
| Input | Project id, lazy bool |
| Return | List<Thread> |
| Beschreibung | Es werden alle Threads mit der als Input überlieferten ProjectOfferId aus der Datenbank abgefragt, die nicht "hidden" sind. Wenn der boolean lazy gesetzt ist, wird lediglich die Id des Threads übergeben, andernfalls wird auch der Titel und die zugehörigen Posts übergeben. |

|  |  |
| --- | --- |
| Methodenname | public Thread getThreadById(int id) |
| Input | Thread Id |
| Return | Thread |
| Beschreibung | Es wird der Thread mit der überlieferten Thread Id aus der Datenbank abgefragt, die nicht "hidden" sind. Der Thread wird mit Titel und zugehörigen Posts übergeben. |

|  |  |
| --- | --- |
| Methodenname | public int addThread(Thread thread) |
| Input | Thread Objekt |
| Return | key als int |
| Beschreibung | Schreibt einen als Input übergebenen Thread in die DB. |

|  |  |
| --- | --- |
| Methodenname | public void editThread(Thread thread) |
| Input | zu änderndes Thread Objekt |
| Return | - |
| Beschreibung | Ändert (Update) den Thread der als Input übergeben wurde. |

|  |  |
| --- | --- |
| Methodenname | public void deleteThreadById(int id) |
| Input | Thread Id |
| Return | - |
| Beschreibung | Ändert (Update) das "hidden"-Attribut des Threads mit der Input Id (hidden = 1) |

6.MySQLPostPersistence.java (LARS)

|  |  |
| --- | --- |
| Methodenname | public Post getPostById(int id) |
| Input | Post Id |
| Return | Post Objekt |
| Beschreibung | Liefert das Postobjekt mit der Input Id. Es werden auch gelöschte (hidden) Posts geliefert, allerdings ohne Post Attachments und User Zugehörigkeit |

|  |  |
| --- | --- |
| Methodenname | public List<Post> getAllPostsByThreadId(int id, boolean lazy) |
| Input | Thread Id, lazy bool |
| Return | Liste der Post Objekte |
| Beschreibung | Liefert eine Liste der Posts mit der Input Id, falls lazy gesetzt ist wird nur die Id der Posts überliefert. Ist lazy nicht gesetzt wird auf hidden geprüft und ggf nur Id und timestamp oder zugehörige PostAttachments und User Zugehörigkeit überliefert. |

|  |  |
| --- | --- |
| Methodenname | public int addPost(Post post) |
| Input | Post Objekt |
| Return | key als int |
| Beschreibung | Schreibt einen als Input übergeben Post in die DB. |

|  |  |
| --- | --- |
| Methodenname | public void editPost(Post post) |
| Input | Post Objekt |
| Return | - |
| Beschreibung | Ändert (Update) den Thread der als Input übergeben wurde. |

|  |  |
| --- | --- |
| Methodenname | public void deletePostById(int id) |
| Input | Post Id als int |
| Return | - |
| Beschreibung | Ändert (Update) das "hidden" Attribut des Posts mit der Input Id (hidden = 1) |

|  |  |
| --- | --- |
| Methodenname | public int getResearchProjectByPostId(int id) |
| Input | Post Id als int |
| Return | Research Project Id als int |
| Beschreibung | Liefert die Research Project Id zum Post mit der Input Id |

7.MySQLPostAttachementPersistence.java (JOCHEN)

|  |  |
| --- | --- |
| Methodenname | Public PostAttachment getPostAttachmentById(int id) |
| Input | int id |
| Return | post Attachment |
| Beschreibung | Es wird ein Anhang aus der Datenbank zu einer bestimmten Id mit den Parametern (id, post\_id, event\_id) zurückgegeben, deren Status deleted = 0 (nicht gelöscht) ist, ansonsten wird eine Exception geworfen, dass kein Angebot zur angegebenen Id gefunden wurde oder der Vorgang fehlschlug. |

|  |  |
| --- | --- |
| Methodenname | Public List<PostAttachment> getAllPostAttachmentByPostId(int id, boolean lazy) |
| Input | int id, boolean lazy |
| Return | postAttachments |
| Beschreibung | Es werden alle Anhänge bezüglich einer bestimmten post\_id aus der Datenbank zurückgegeben, ansonsten wird eine Exception zurückgegeben, dass keine Anhänge zur post\_id gefunden werden konnten. |

|  |  |
| --- | --- |
| Methodenname | Public int addPostAttachment(PostAttachment postAttachment) |
| Return | key |
| Beschreibung | Hinzufügen eines neuen Anhangs in die Datenbank mit den Parametern (post\_id, event\_id, name), ansonsten wird eine Exception geworfen, dass der Vorgang fehlschlug. |

|  |  |
| --- | --- |
| Methodenname | Public void deleteAttachmentById(int id) |
| Input | int id |
| Beschreibung | Löscht einen Anhang mit bestimmter Id, indem der Parameter deleted = 1 (gelöscht) gesetzt wird, ansonsten wird eine Exception geworfen, dass der Vorgang nicht durchgeführt werden konnte. |

|  |  |
| --- | --- |
| Methodenname | Public void deleteAttachmentByPostId(int id) |
| Input | int id |
| Beschreibung | Löscht alle Anhänge bezügliche einer bestimmten post\_Id, indem der Parameter deleted = 1 (gelöscht) gesetzt wird, ansonsten wird eine Exception geworfen, dass nicht alle Anhänge zur angegebenen post\_id gelöscht werden konnten. |

|  |  |
| --- | --- |
| Methodenname | Public void setContentById(int id, byte[] bytes) |
| Input | int id |
| Beschreibung | Fügt einem Anhang mit bestimmter Id einen neuen Inhalt hinzu, ansonsten wird eine Exception geworfen, dass der Vorgang fehlschlug oder die länge des Inhalts überschritten wurde. |

|  |  |
| --- | --- |
| Methodenname | Public InputStream getContentById(int id) |
| Input | int id |
| Return | content |
| Beschreibung | Gibt den Inhalt eines Anhangs mit bestimmter Id zurück, ansonsten wird eine Exception geworfen, dass kein Inhalt verfügbar ist oder die angegebene Anhangs-Id nicht gefunden wurde. |

8.MySQLResearchProjectPaperPersistence.java (edit-Methode) (LARS)

|  |  |
| --- | --- |
| Methodenname | public void editProject(ResearchProjectPaper paper) |
| Input | ResearchProjectPaper Objekt |
| Return | - |
| Beschreibung | Es wird überprüft ob die paperId bereits als projectId zu einem Offer oder Paper Objekt gehört. Zuerst wird das research\_project in der DB geändert (Update). Existiert ein Paper mit der paperId wird dann das research\_project\_paper in der DB geändert (Update). Existiert ein Offer mit der paperId ??????? |

# REST

## StudentREST.java

* Die Klasse StudentRest.java bildet prinzipiell Methoden ab, die überprüfen ob ein Schülerobjekt verfügbar, bzw. Eine Rolle hinzufügbar, editierbar, löschbar ist (auch welche Rollen dem Objekt zugeordnet wurden).
* Der Rückgabewert ist in allen Methoden ok, serverError oder eine Exception.
* Erbt von AccessPermissionEndpoint

## 

|  |  |
| --- | --- |
| Methodenname | public LazyObject **getLazyObject**(**int** id) |
| Input | Id |
| Return | persistence.getRoleById(id) |
| Beschreibung |  |

|  |  |
| --- | --- |
| Methodenname | public Response **getById**(**@PathParam**("id") **int** id)  Response: Defines the contract between a returned instance and the runtime when an application needs to provide meta-data to the runtime. |
| Input | @PathParam("id") int id |
| Return | **return** Response.ok(student).build();  Rückgabewert wenn kein Fehler auftritt (try/catch) und das "student" Objekt ungleich null.  **return** Response.status(Status.NOT\_FOUND).build();  Rückgabewert wenn kein Fehler auftritt (try/catch) und das "student" Objekt nicht gefunden wurde.  **return** Response.serverError().entity(e.getMessage()).build();  Rückgabewert bei Exception |
| Beschreibung | Von der Klasse "Student" wird ein Objekt erzeugt und die entsprechende Rolle - abhängig von der übergebenen ID – darin gespeichert.  Student student = persistence.**getRoleById**(id);  Je nachdem ob eine Exception auftritt oder das Objekt nicht gefunden wurde, wird der entsprechende return wert zurückgegeben. |

|  |  |
| --- | --- |
| Methodenname | public Response **add**(Student lazyObject) |
| Input | lazyObject |
| Return | **return** Response.**ok**(persistence.**getRoleById**(id)).**build**();  Rückgabewert wenn keine Exception auftritt.  **return** Response.**serverError**().**entity**(e.**getMessage**()).**build**();  Rückgabewert bei Exception  Logger.error("An error occured while adding a Studentrole"); |
| Beschreibung | Fügt dem entsprechenden Schülerobjekt eine Role hinzu und speichert die id in einer lokalen variable ab. Abhängig davon ob eine Exception auftritt, werden unterschiedliche Rückgabewerte aufgerufen. |

|  |  |
| --- | --- |
| Methodenname | public Response **edit**(Student lazyObject) |
| Input | lazyObject |
| Return |  |
| Beschreibung | Die Methode persistence.**editRole**(lazyObject)  wird aufgerufen, falls keine Exception auftritt. Abhängig davon ob eine Exception auftritt, werden unterschiedliche Rückgabewerte aufgerufen. |

|  |  |
| --- | --- |
| Methodenname | public Response **delete**(**@PathParam**("id") **int** id) |
| Input | lazyObject |
| Return | **return** Response.ok(persistence.getRoleById(lazyObject.getID()).build();  **return** Response.serverError().entity(e.getMessage()).build(); |
| Beschreibung | Löscht eine Rolle des übergebenen Schülerobjekts durch den entsprechenden Methodenaufruf. Abhängig davon ob eine Exception auftritt, werden unterschiedliche Rückgabewerte aufgerufen. |

## TeacherRest.java

* Die Klasse TeacherRest.java bildet prinzipiell Methoden ab, die überprüfen ob ein Teacherobjekt verfügbar, bzw. Eine Rolle hinzufügbar, editierbar, löschbar ist (auch welche Rollen dem Objekt zugeordnet wurden)
* Der Rückgabewert ist in allen Methoden ok, serverError oder eine Exception.
* Erbt von AccessPermissionEndpoint

|  |  |
| --- | --- |
| Methodenname | public LazyObject **getLazyObject**(int id) |
| Input | id |
| Return | Return persistence.getRoleById(id) |
| Beschreibung |  |

|  |  |
| --- | --- |
| Methodenname | public Response **getById**(@PathParam("id") int id) |
| Input | @PathParam("id") int id |
| Return | **return** Response.ok(teacher).build();  Rückgabewert wenn kein Fehler auftritt (try/catch) und das "teacher" Objekt ungleich null.  **return** Response.status(Status.NOT\_FOUND).build();  Rückgabewert wenn kein Fehler auftritt (try/catch) und das "student" Objekt nicht gefunden wurde.  **return** Response.serverError().entity(e.getMessage()).build();  Rückgabewert bei Exception |
| Beschreibung | Von der Klasse "Teacher" wird ein Objekt erzeugt und die entsprechende Rolle - abhängig von der übergebenen ID – darin gespeichert.  Student student = persistence.**getRoleById**(id);  Je nachdem ob eine Exception auftritt oder das Objekt nicht gefunden wurde, wird der entsprechende return wert zurückgegeben. |

|  |  |
| --- | --- |
| Methodenname | public Response **add**(Teacher lazyObject) |
| Input | lazyObject |
| Return | **return** Response.ok(persistence.getRoleById(id)).build();  Rückgabewert wenn keine Exception auftritt.  **return** Response.serverError().entity(e.getMessage()).build();  Rückgabewert bei Exception  Logger.error("An error occured while adding a Teacherrole"); |
| Beschreibung | Fügt dem entsprechenden Schülerobjekt eine Role hinzu und speichert die id in einer lokalen variable ab. Abhängig davon ob eine Exception auftritt, werden unterschiedliche Rückgabewerte aufgerufen. |

|  |  |
| --- | --- |
| Methodenname | public Response **edit**(Teacher lazyObject) |
| Input | lazyObject |
| Return | **return** Response.ok(persistence.getRoleById(lazyObject.getId())).build();  Rückgabewert wenn keine Exception auftritt  **return** Response.serverError().entity(e.getMessage()).build();  Rückgabewert bei Exception |
| Beschreibung | Die Methode **edit** ruft die Methode persistence.**editRole**(lazyObject) zum editieren einer Role auf. Abhängig davon ob eine Exception auftritt wird ein entsprechender return wert gebildet. |

|  |  |
| --- | --- |
| Methodenname | public Response **delete**(@PathParam("id") int id) |
| Input | @PathParam("id") int id |
| Return | **return** Response.ok("True").build;  Rückgabewert wenn die Rolle gelöscht wurde.  **return** Response.serverError().enity(e.getMessage()).build();  Rückgabewert bei Exception |
| Beschreibung | Die Methode löscht eine Rolle abhängig von der übergebenen Teacher id. Abhängig davon ob eine Exception auftritt wird ein entsprechender return Wert generiert. |

## ThreadRest.java

|  |  |
| --- | --- |
| Methodenname | Public LazyObject getLazyObject(int id) |
| Input | Int id |
| Return | Return persistence.getThreadbyId(id); |
| Beschreibung |  |

|  |  |
| --- | --- |
| Methodenname | Public Response getById(@PathParam("id") int id); |
| Input | @PathParam("id") int id |
| Return | Return Response.ok(thread).build();  Rückgabewert wenn Thread != null  Return Resonse.status(Satus.NOT\_FOUND).build();  Rückgabewert wenn Thread nicht gefunden wurde.  Return Response.serverError(). Entity(e.getMessage()).build();  Rückgabewert wenn eine Exception auftritt. |
| Beschreibung | Speichert einen Thread abhängig von der ID in einem ThreadObjekt und erzeugt abhängig von einer eventuell auftretenden Exception, einen entsprechenden Rückgabewert. |

|  |  |
| --- | --- |
| Methodenname | Public Response addThread (thread) |
| Input | thread |
| Return | Return Response.ok(createdThread).build()  Rückgabewert wenn Thread erzeugt wurde.  Return respronse.serverError().entity(e.getMessage()).build();  Rückgabewert bei Exception |
| Beschreibung | Erstellt einen neuen Thread. |

|  |  |
| --- | --- |
| Methodenname | Public Response edit(Thread thread) |
| Input | Thread thread |
| Return | Return Response.status(Status.FORBIDDEN).build(); |
| Beschreibung | Erzeugt einen Rückgabewert, dass das Editieren von Threads nicht erlaubt ist. |

|  |  |
| --- | --- |
| Methodenname | Public Response delete(@PathParam("id") int id) |
| Input | @PathParam("id") int id |
| Return | Return Response.ok("true").build();  Rückgabewert wenn KEINE Exception auftritt  Return Response.serverError(). Entity(e.getMessage()).build();  Rückgabewert wenn eine Exception auftritt. |
| Beschreibung | Löscht einen Thread abhängig von der übergebenen id |

## 

## ResearchProjectOfferRest.java

|  |  |
| --- | --- |
| Methodenname | Public LazyObject getLazyObject(int id) |
| Input | Int id |
| Return | Return persistence.getProjectById(id); |
| Beschreibung |  |

|  |  |
| --- | --- |
| Methodenname | Public Response getAll() |
| Input | - |
| Return | Return Response.ok(allOffers).build();  Rückgabewert wenn alle Offers im Listobject "allOffers" gespeichert wurden.  Return Response.serverErroer().entity(e.getMessage()).build();  Rückgabewert bei Exception |
| Beschreibung | Speichert alle Angebote in einer Liste. Rückgabewert abhängig davon ob eine Exception auftritt oder nicht. |

|  |  |
| --- | --- |
| Methodenname | Public Response getByID(@PathPram("id") int id) |
| Input | @PathPram("id") int id |
| Return | return Response.ok(offer).build();  Rückgabewert wenn das Angebot im Objekt gespeichert wurde.  return Response.status(Status.NOT\_Found).build();  Rückgabewert wenn kein Angebot mit der übergebenen id gefunden wurde.  Return Response.serverError().entity(e.getMessage()).build();  Rückgabewert bei Exception. |
| Beschreibung | Speichert ein Angebot abhängig von der id im Objekt "offer". |

|  |  |
| --- | --- |
| Methodenname | public Response getByUuuid(@PathParam("uuid") String id) |
| Input | @PathParam("uuid") String id |
| Return | Return Response.ok(currPaper).build();  Rückgabewert wenn Project im Objekt "currPaper" gespeichert wurde.  Return Response.server().entity(e.getMessage()).build();  Rückgabewert bei Exception. |
| Beschreibung | Speichert ein Projekt im Objekt "currPaper" abhängig von der übergebenen id.  Aufruf der Methode persistence.getProjectByUuid(id); |

|  |  |
| --- | --- |
| Methodenname | Public Response add(ResearchProjectOffer off) |
| Input | ResearchProjectOffer offer |
| Return | Return Response.ok(createdOffer).build();  Rückgabewert wenn das neue Projekt erfolgreich angelegt wurde.  Return Response.serverError().entity(e.getMessage()).build();  Rückgabewert bei Exception. |
| Beschreibung | Erzeugt ein neues Projekt und speichert dieses im Objekt "createdOffer". |

|  |  |
| --- | --- |
| Methodenname | Public Response edit(ResearchProjectOffer offer) |
| Input | ResearchProjectOffer offer |
| Return | Return Response.ok(persistence.getProjectById(offer.getId())).buid();  Rückgabewert wenn edit Methode erforlgreich aufgerufen wird.  Return Response.serverErroer().entity(e.getMessage()).build();  Rückgabewert wenn Exception auftritt. |
| Beschreibung | Ruft die Methode zum editieren von Angeboten auf und erzeugt einen entsprechenden Rückgabewert. |

|  |  |
| --- | --- |
| Methodenname | Public Response delete(@PathParam("id") int id) |
| Input |  |
| Return |  |
| Beschreibung |  |

## ResearchProjectPaperREST.java

* Die Klasse ResearchProjectPaperREST.java enthält Methoden, die auf Research Project Papers (RPPs) zurückgreifen.
* Erbt von AccessPermissionEndpoint

|  |  |
| --- | --- |
| Methodenname | public Response edit(ResearchProjectPaper paper) |
| Input | ResearchProjectPaper paper |
| Return | return Response.ok(persistence.getProjectById(paper.getID())).build();  Rückgabewert wenn editierbares RPP vorliegt  return Response.serverError().entity(e.getMessage()).build();  Rückgabewert bei Exception |
| Beschreibung | Editiert ein RPP in Abhängigkeit der übergebenen Id. |

## UserREST.java

* Die Klasse UserREST.java enthält Methoden, die sich auf *User* beschränken.
* Erbt von AccessPermissionEndpoint

|  |  |
| --- | --- |
| Methodenname | public Response getAll() { |
| Input | - |
| Return | return Response.ok(users).build();  Rückgabewert wenn alle User in List<User> abgespeichert wurden  return Response.status(Status.NOT\_FOUND).build();  Rückgabewert wenn fullUsers = null  return Response.serverError().entity(e.getMessage()).build();  Rückgabewert bei Exception |
| Beschreibung | Gibt alle User mit Angabe zu Id, Login, Firstname, Lastname, Title, Email und Roles zurück. |

|  |  |
| --- | --- |
| Methodenname | public Response getById(@PathParam("id") int id) { |
| Input | @PathParam("id") int id |
| Return | Return Response.ok(user).build();  Rückgabewert wenn KEINE Exception auftritt  return Response.status(Status.NOT\_FOUND).build();  Rückgabewert wenn user id = null  Return Response.serverError(). Entity(e.getMessage()).build();  Rückgabewert wenn eine Exception auftritt. |
| Beschreibung | Gibt einen User abhängig von der ID zurück. Je nach AccessPermission bekommt das Objekt die Parameter Id, Login, Firstname, Lastname, Title (case PARTIALLY) und zusätzlich EMAIL (case ALLOWED) mitgeliefert. |

|  |  |
| --- | --- |
| Methodenname | public Response add(User user) |
| Input | User user |
| Return | return Response.ok(persistence.getUserById(id)).build();  Rückgabewert wenn User erzeugt wurde.  Return respronse.serverError().entity(e.getMessage()).build();  Rückgabewert bei Exception |
| Beschreibung | Fügt einen User mit einer id (Integer) hinzu. |

|  |  |
| --- | --- |
| Methodenname | public Response edit(User user) |
| Input | User user |
| Return | return Response.status(Status.CONFLICT).build();  Rückgabewert falls Konflikt mit anderem Login auftritt  return Response.ok(persistence.getUserById(user.getId())).build();  Rückgabewert wenn User editiert wurde.  Return response.serverError().entity(e.getMessage()).build();  Rückgabewert bei Exception |
| Beschreibung | Editiert einen User in Abhängigkeit der id. Vorgeschaltet ist eine Überprüfung des Logins. |

|  |  |
| --- | --- |
| Methodenname | public Response delete(User user) |
| Input | @PathParam("id") int id) |
| Return | return Response.ok("true").build();  Rückgabewert wenn User gelöscht wurde.  Return respronse.serverError().entity(e.getMessage()).build();  Rückgabewert bei Exception |
| Beschreibung | Löscht einen User in Abhängigkeit der übergebenen id. |

|  |  |
| --- | --- |
| Methodenname | public Response changePassword(PasswordChangeRequest passwordChangeRequest) |
| Input | PasswordChangeRequest passwordChangeRequest |
| Return | return Response.ok(passwordChangeRequest).build();  Rückgabewert wenn Passwort geändert wurde.  Return respronse.serverError().entity(e.getMessage()).build();  Rückgabewert bei Exception |
| Beschreibung | Ändert das Passwort eines Users in Abhängigkeit seiner Id. |

## PostREST.java

* Die Klasse PostREST.java enthält Methoden, die die Behandlung von Posts beschreiben.
* Erbt von AccessPermissionEndpoint

|  |  |
| --- | --- |
| Methodenname | public Response getById(@PathParam("id") int id) |
| Input | @PathParam("id") int id |
| Return | return Response.ok(post).build();  Rückgabewert wenn Post mit Id vorhanden  return Response.status(Status.NOT\_FOUND).build();  Rückgabewert wenn post id = null  return Response.serverError().entity(e.getMessage()).build();  Rückgabewert bei Exception |
| Beschreibung | Gibt einen Post in Abhängigkeit der übergebenen id zurück. |

|  |  |
| --- | --- |
| Methodenname | public Response add(Post post) |
| Input | Post post |
| Return | return Response.ok(persistence.getPostById(id)).build();  Rückgabewert wenn Post erstellt wurde.  return Response.serverError().entity(e.getMessage()).build();  Rückgabewert bei Exception |
| Beschreibung | Fügt einen Post mit einer id (Integer) hinzu. |

|  |  |
| --- | --- |
| Methodenname | public Response edit(Post post) |
| Input | Post post |
| Return | return Response.serverError().entity(re.getMessage()).build();  Rückgabewert bei Exception  return Response.ok(persistence.getPostById(post.getId())).build();  Rückgabewert wenn "hidden" flag auf true gesetzt wurde.  return Response.status(Status.FORBIDDEN).build();  Rückgabewert bei Versuch, Post zu ändern (ausschließlich "hidden"-flag) |
| Beschreibung | Setzt bei Bedarf den "hidden"-Flag des Posts, weitere Änderungen an Posts sind nicht möglich. |

|  |  |
| --- | --- |
| Methodenname | public Response delete(@PathParam("id") int id) |
| Input | @PathParam("id") int id |
| Return | return Response.status(Status.FORBIDDEN).build(); |
| Beschreibung | Posts können nicht gelöscht werden (nur Setzen des "hidden"-Flags erlaubt) |

## PostAttachmentREST.java

* Die Klasse PostAttachmentREST.java enthält Methoden, die sich Post-Anhänge beschäftigt.
* Erbt von AccessPermissionEndpoint

|  |  |
| --- | --- |
| Methodenname | public Response getById(@PathParam("id") int id) |
| Input | @PathParam("id") int id |
| Return | return Response.ok(postAttachment).build();  Rückgabewert wenn Attachment vorhanden  return Response.status(Status.NOT\_FOUND).build();  Rückgabewert wenn postAttachment id = null  return Response.serverError().build();  Rückgabewert bei Exception |
| Beschreibung | Gibt postAttachment in Abhängigkeit der übergebenen id zurück. |

|  |  |
| --- | --- |
| Methodenname | public Response add(PostAttachment lazyObject) |
| Input | PostAttachment lazyObject |
| Return | return Response.ok(persistence.getPostAttachmentById(id)).build();  Rückgabewert wenn Attachment hinzugefügt wurde  return Response.serverError().build();  Rückgabewert bei Exception |
| Beschreibung | Fügt ein PostAttachment mit einer id (Integer) hinzu. |

|  |  |
| --- | --- |
| Methodenname | public Response edit(PostAttachment lazyObject) |
| Input | PostAttachment lazyObject |
| Return | return null; |
| Beschreibung | Funktion deaktiviert |

|  |  |
| --- | --- |
| Methodenname | public Response delete(int id) |
| Input | int id |
| Return | return null; |
| Beschreibung | Funktion deaktiviert |

|  |  |
| --- | --- |
| Methodenname | public Response setContent(@PathParam("id") int id, List<org.apache.cxf.jaxrs.ext.multipart.Attachment> attachments) |
| Input | @PathParam("id") int id, List<org.apache.cxf.jaxrs.ext.multipart.Attachment> attachments |
| Return | return Response.ok().build();  Rückgabewert wenn Content gesetzt wurde  return Response.status(413).build();  Rückgabewert wenn Attachment zu groß ist und es gelöscht wurde  return Response.serverError().build();  Rückgabewert bei Exception |
| Beschreibung | Schreibt den Inputstream in ein Byte-Array ("bytes") und erstellt in Abhängigkeit der übergebenen id einen Content auf Basis des Array-Inhaltes. |

|  |  |
| --- | --- |
| Methodenname | public Response getContent(@PathParam("id") int id) |
| Input | @PathParam("id") int id |
| Return | return Response.ok(is)  .header("Content-Disposition", "attachment; filename=\"" + postAttachment.getName() + "\"")  .build();  Rückgabewert wenn Content abgerufen wurde.  return Response.serverError().build();  Rückgabewert bei Exception |
| Beschreibung | Sucht einen Content anhand der übergebenen id und erstellt einen Header um die Content-Datei per http/html zu verschicken. Die Datei liegt als InputStream vor. |