

# A5 - Web Services

Daniel Sánchez  
Edgar Moreno

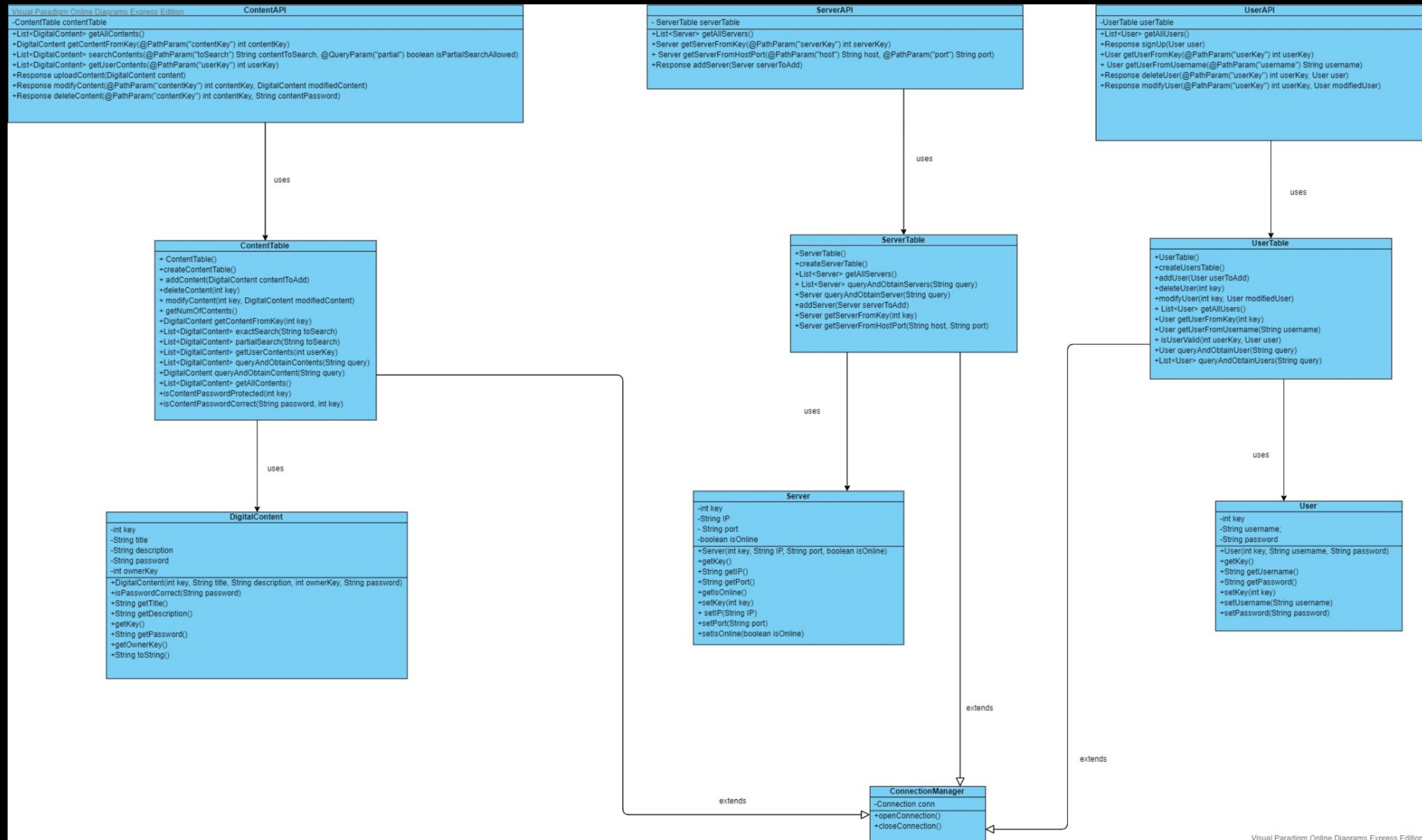
# entities

contents

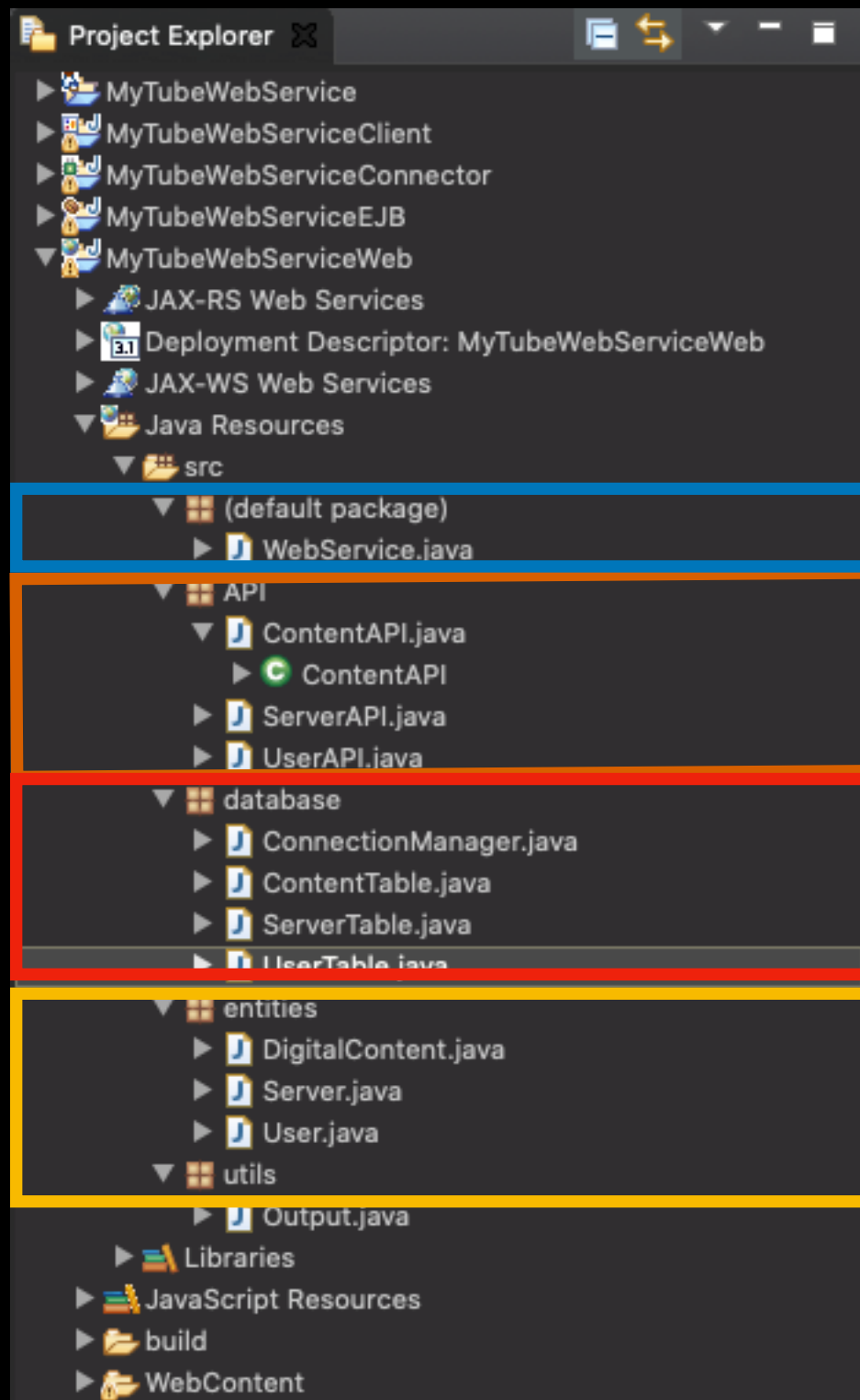
users

servers

# uml



# project structure



## WebService.java

defines the `@ApplicationPath (/api)`

## API related classes

define the paths and operations for the different entities, triggered by Java clients or direct access from browser

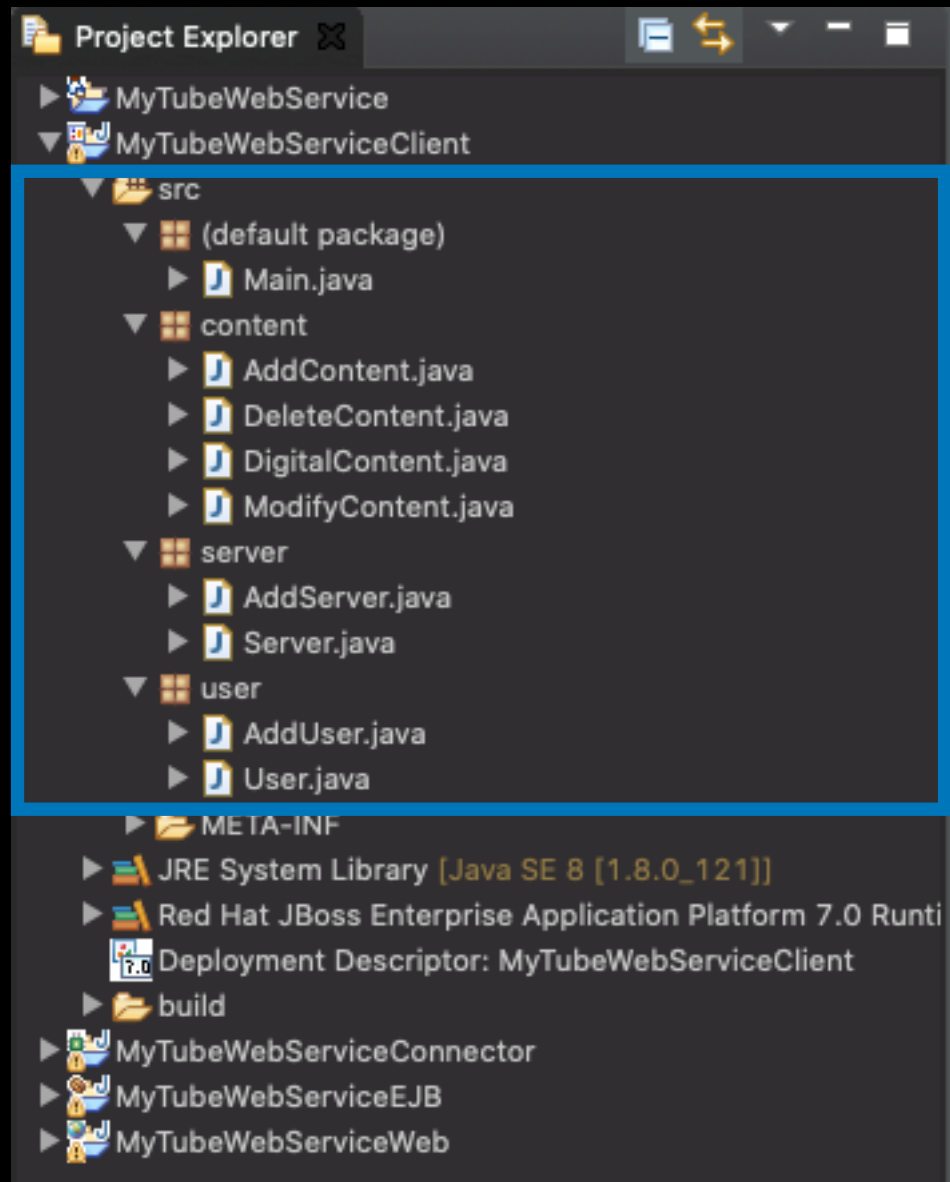
## Database related classes

define the operations for the different databases, triggered by the API related classes

## Auxiliar classes

define operations for packing, unpacking and formatting purposes

# project structure



**Different clients for PUT, POST and DELETE operations**

GET operations can be triggered directly from the browser or using tools like curl

# endpoints > contents

endpoint	description	returning value
GET /content	gets all contents	“application/json”
GET /content/{contentKey}	gets content from key	“application/json”
GET /content/user/{userKey}	gets user’s contents	“application/json”
GET /content/server/{serverKey}	gets server’s contents	“application/json”
GET /content/search/{toSearch}?partial=False	exact/partial search on content’s description and password. partial is optional and False by default	“application/json”

# endpoints > contents

endpoint	description	returning value
<b>POST /content</b>	creates content	201 (if added in database), 409 (if content's title already exists) or 500 (if can't be added)
<b>PUT /content/{contentKey}</b>	modifies content (must provide the correct content password)	201 (if modified), 401 (if content's password is incorrect), 409 (if content does not exist or 500 (if can't be modified)
<b>DELETE /content/{contentKey}</b>	deletes content (must provide the correct content password)	201 (if deleted), 401 (if content's password is incorrect), 409 (if content does not exist or 500 (if can't be deleted)

# endpoints > users

endpoint	description	returning value
GET /user	gets all users	"application/json"
GET /user/{userKey}	gets user from key	"application/json"
GET /user/username/{username}	gets user from username	"application/json"
POST /user	creates user	201 (if added in database), 409 (if username is taken) or 500 (if can't be added)



# endpoints > servers

endpoint	description	returning value
GET /server	gets all servers	“application/json”
GET /server/{serverKey}	gets server from key	“application/json”
GET /server/host/{host}/port/{port}	gets server from host and port	“application/json”
POST /server	creates server	201 (if added in database) or 500 (if can't be added)

