A5 - Web Services

Daniel Sánchez
Edgar Moreno

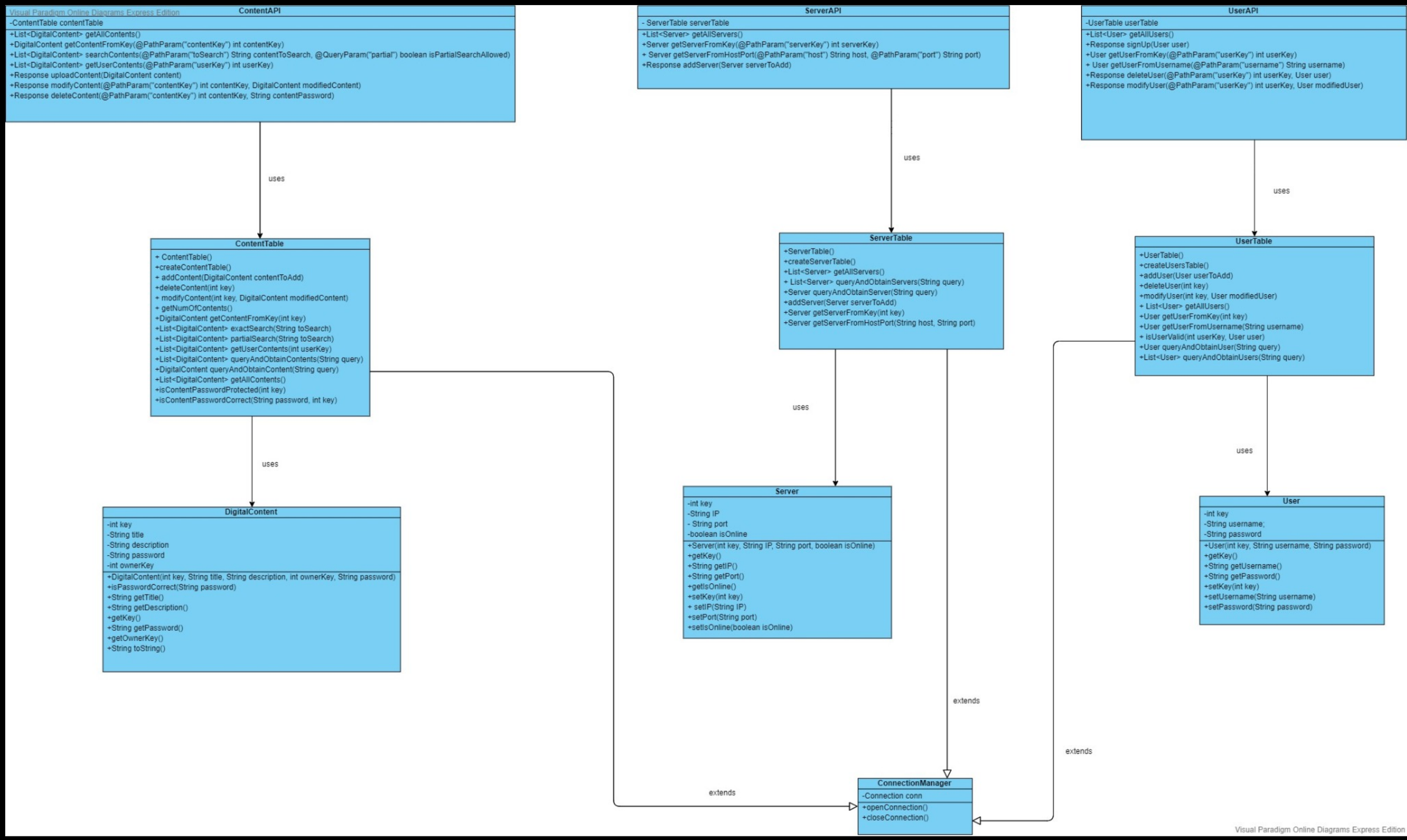
entities

contents

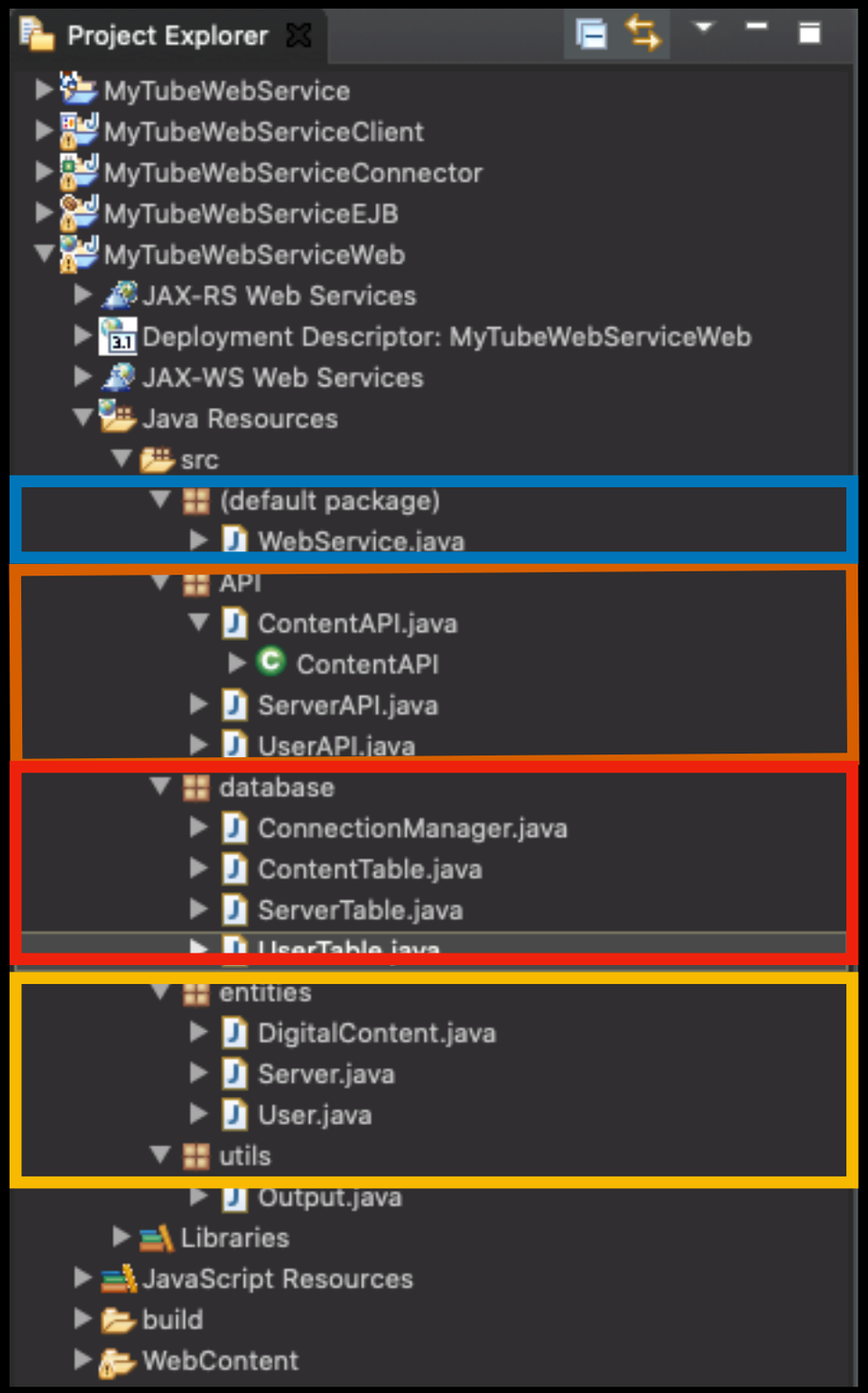
users

servers

uml


Visual Paradigm Online Diagrams Express Edition
ContentAPI
-ContentTable contentTable
+List<DigitalContent> getAllContents()
+DigitalContent getContentFromKey(@PathParam("contentKey") int contentKey)
+List<DigitalContent> searchContents(@PathParam("toSearch") String contentToSearch, @QueryParam("partial") boolean isPartialSearchAllowed)
+List<DigitalContent> getUserContents(@PathParam("userKey") int userKey)
+Response uploadContent(DigitalContent content)
+Response modifyContent(@PathParam("contentKey") int contentKey, DigitalContent modifiedContent)
+Response deleteContent(@PathParam("contentKey") int contentKey, String contentPassword)
uses
ContentTable
+ ContentTable()
+createContentTable()
+ addContent(DigitalContent contentToAdd)
+deleteContent(int key)
+ modifyContent(int key, DigitalContent modifiedContent)
+ getNumOfContents()
+DigitalContent getContentFromKey(int key)
+List<DigitalContent> exactSearch(String toSearch)
+List<DigitalContent> partialSearch(String toSearch)
+List<DigitalContent> getUserContents(int userKey)
+List<DigitalContent> queryAndObtainContents(String query)
+DigitalContent queryAndObtainContent(String query)
+List<DigitalContent> getAllContents()
+isContentPasswordProtected(int key)
+isContentPasswordCorrect(String password, int key)
uses
DigitalContent
-int key
-String title
-String description
-String password
-int ownerKey
+DigitalContent(int key, String title, String description, int ownerKey, String password)
+isPasswordCorrect(String password)
+String getTitle()
+String getDescription()
+getKey()
+String getPassword()
+getOwnerKey()
+String toString()
ServerAPI
- ServerTable serverTable
+List<Server> getAllServers()
+Server getServerFromKey(@PathParam("serverKey") int serverKey)
+ Server getServerFromHostPort(@PathParam("host") String host, @PathParam("port") String port)
+Response addServer(Server serverToAdd)
uses
ServerTable
+ServerTable()
+createServerTable()
+List<Server> getAllServers()
+ List<Server> queryAndObtainServers(String query)
+Server queryAndObtainServer(String query)
+addServer(Server serverToAdd)
+Server getServerFromKey(int key)
+Server getServerFromHostPort(String host, String port)
uses
Server
-int key
-String IP
- String port
-boolean isOnline
+Server(int key, String IP, String port, boolean isOnline)
+getKey()
+String getIP()
+String getPort()
+getIsOnline()
+setKey(int key)
+ setIP(String IP)
+setPort(String port)
+setIsOnline(boolean isOnline)
UserAPI
-UserTable userTable
+List<User> getAllUsers()
+Response signUp(User user)
+User getUserFromKey(@PathParam("userKey") int userKey)
+ User getUserFromUsername(@PathParam("username") String username)
+Response deleteUser(@PathParam("userKey") int userKey, User user)
+Response modifyUser(@PathParam("userKey") int userKey, User modifiedUser)
uses
UserTable
+UserTable()
+createUsersTable()
+addUser(User userToAdd)
+deleteUser(int key)
+modifyUser(int key, User modifiedUser)
+ List<User> getAllUsers()
+User getUserFromKey(int key)
+User getUserFromUsername(String username)
+ isUserValid(int userKey, User user)
+User queryAndObtainUser(String query)
+List<User> queryAndObtainUsers(String query)
uses
User
-int key
-String username;
-String password
+User(int key, String username, String password)
+getKey()
+String getUsername()
+String getPassword()
+setKey(int key)
+setUsername(String username)
+setPassword(String password)
extends
extends
extends
ConnectionManager
-Connection conn
+openConnection()
+closeConnection()
Visual Paradigm Online Diagrams Express Edition

project structure



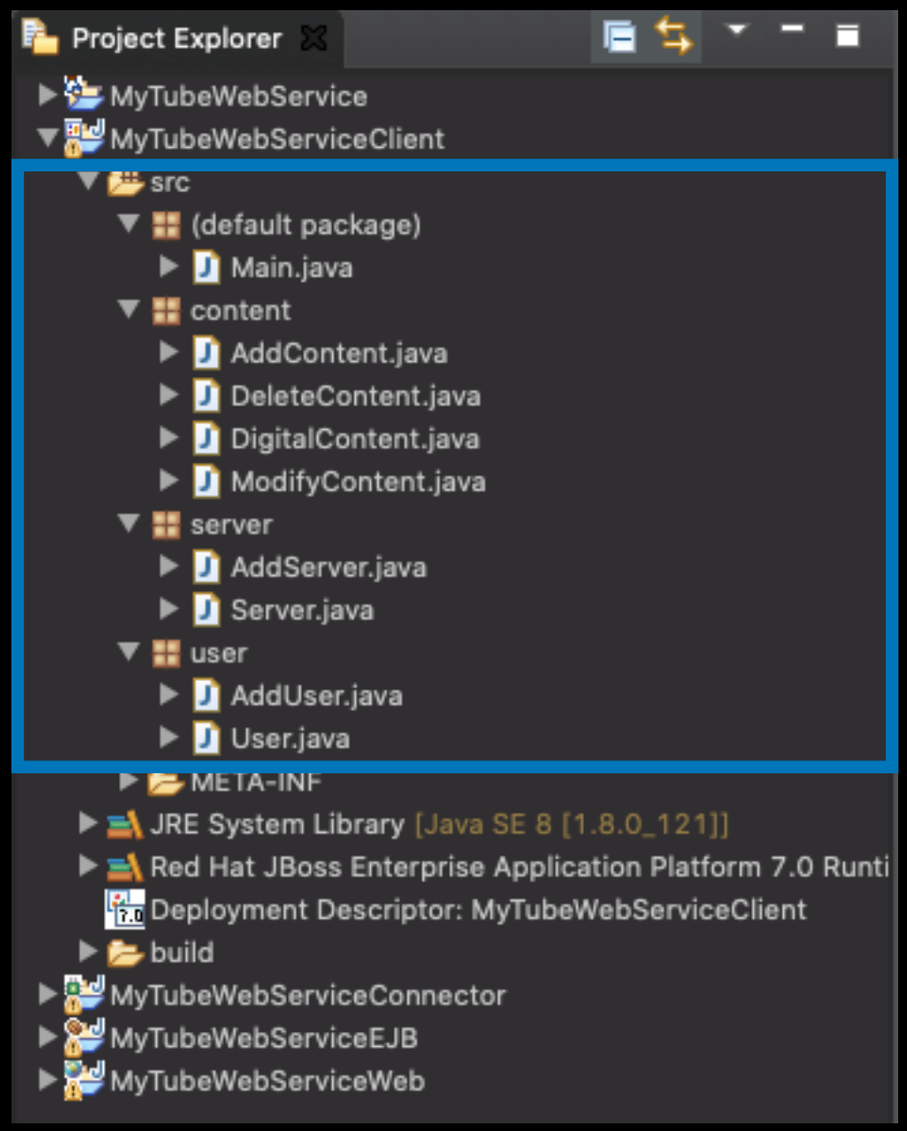
WebService.java
defines the @ApplicationPath (/api)

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
POST /content	creates content	201 (if added in database), 409 (if content's title already exists) or 500 (if can't be added)
PUT /content/{contentKey}	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)
DELETE /content/ {contentKey}	deletes content (must provide the correct content password)	200 (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), 409 (if server with same host:port already exists) or 500 (if can’t be added)