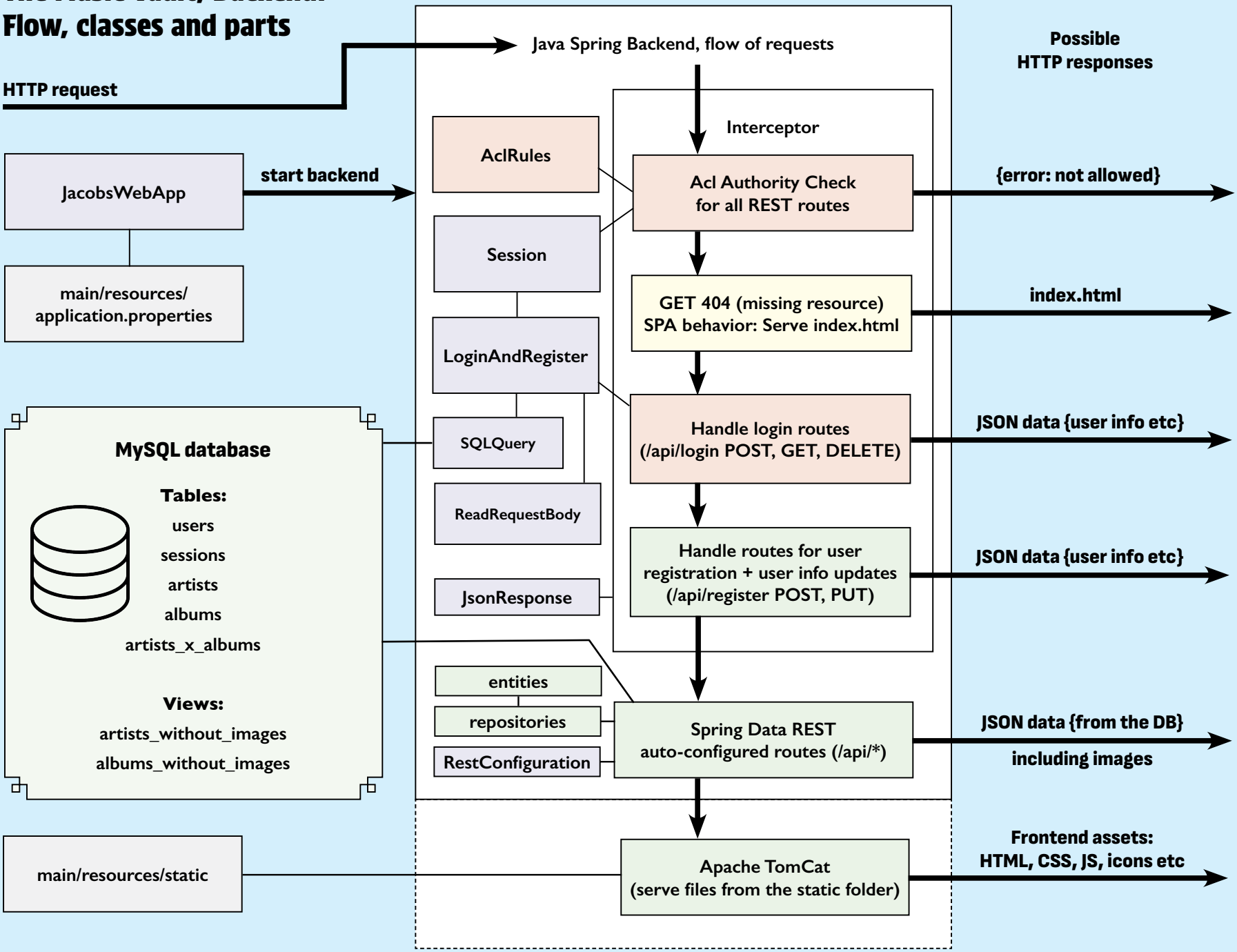
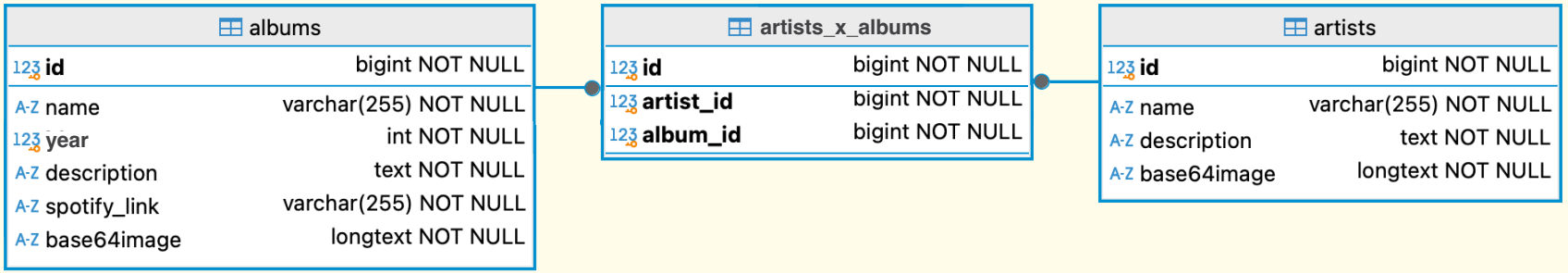
The Music Vault, Backend: Flow, classes and parts

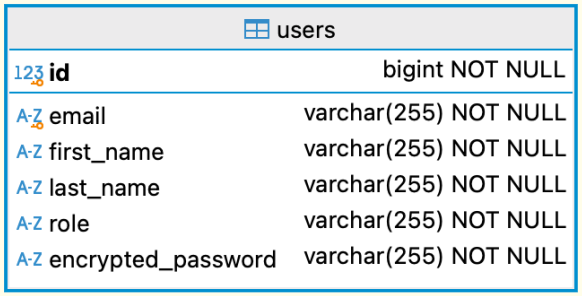


The Music Vault, MySQL database: ER Diagram

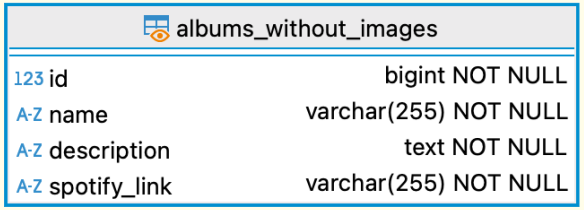
The only relation we have is a many-to-many relationship between albums and artists:

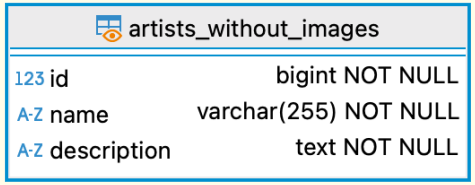
- An album is made by one or several artists, an artist can make one to many albums
- We solve this many-to-many relation in the classic SQL way, by introducing a join table, **artistsXalbums**.





Users are stored with BCrypt-encrypted passwords in the table **users**. Data for logged in users are stored in the table **sessions**, where the session_id is a random hash used as our cookie value.

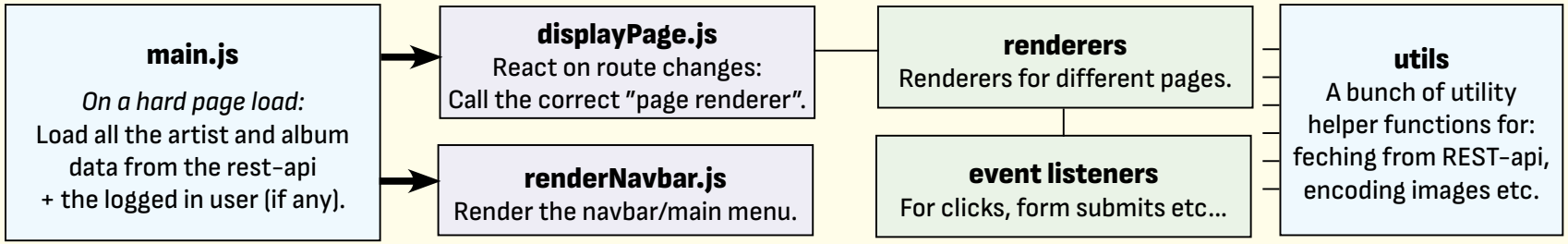




There are two views in the data-base, showing the data from the **artists** and **albums** tables without the base64 encrypted images.

The Music Vault: Quick overview

The frontend is an SPA built with "vanilla JS" (JavaScript without any frontend framework). Two external JS libraries are used: The JS part of the Bootstrap CSS library and Marked.js (*conversion of markdown to html*). The Bootstrap CSS library is used to accomplish a modern responsive layout in conjunction with my own CSS.



SPA (Single Page Application): Frontend routing

The frontend is an SPA, which takes care of the routing of a number of different GET routes, via JavaScript running in the browser, without doing any hard page reloads between route changes.

Menu item / Link	Shown for	GET route	Description
Artists	everyone	/	List of artists
Albums	everyone	/albums	List of albums
Add an artist	logged in users	/add-artist	Form for adding an artist
Add an album	logged in users	/add-album	Form for adding an album
About	everyone	/about	Info about the application
Register	visitors (not logged in)	/register	Form for registering a user
[name of logged in user]	logged in users	/register	Form for updating user info
[link from each artist in list]	everyone	/artists-info/:id	Details about an artist
[link from each album in list]	everyone	/album-info/:id	Details about an album
[link from each artist in list]	logged in users	/edit-artist/:id	Form for editing artist info
[link from each album in list]	logged in usrs	/edit-album/:id	Form for editing album info

The Music Vault: REST API Routes

Request method	**Route**	**Access control** *(by user roles)*
POST	/api/register	visitor
PUT	/api/register	user
POST	/api/login	visitor
GET	/api/login	visitor, user
DELETE	/api/login	user
POST	/api/artists	user
GET	/api/artists	visitor, user
GET	/api/artists/:id	visitor, user
PUT	/api/artists/:id	user
DELETE	/api/artists/:id	user
GET	/api/artistsWithoutImages	visitor, user
GET	/api/artistsWithoutImages/:id	visitor, user
POST	/api/albums	user
GET	/api/albums	visitor, user
GET	/api/albums/:id	visitor, user
PUT	/api/albums/:id	user
DELETE	/api/albums/:id	user
GET	/api/albumsWithoutImages	visitor, user
GET	/api/albumsWithoutImages/:id	visitor, user
GET	/artistXalbums	visitor, user
POST	/artistXalbums	user
DELETE	/artistXalbums/:id	user

The Music Vault: File Statistics, version 0.8.3

Java	SLOC	KB
AclRules.java	41	1.5
AsciiLogo.java	15	0.8
Interceptor.java	116	4.1
JacobsWebApp.java	94	3.6
JsonResponse.java	18	0.6
LoginAndRegister.java	113	4.2
ReadRequestBody.java	49	1.6
RestConfiguration.java	37	1.4
SQLQuery.java	50	1.6
Session.java	62	2.0
SystemOutOnOff.java	23	0.5
entities/Album.java	33	0.7
entities/AlbumWithoutImage.java	27	0.6
entities/Artist.java	28	0.6
entities/ArtistWithoutImage.java	24	0.5
entities/ArtistXAlbum.java	25	0.5
entities/User.java	34	0.7
repositories/AlbumRepository.java	8	0.3
repositories/AlbumWithoutImageRepository.java	8	0.4
repositories/ArtistRepository.java	8	0.3
repositories/ArtistWithoutImageRepository.java	8	0.4
repositories/ArtistXAlbumRepository.java	8	0.4
repositories/UserRepository.java	8	0.3
SUM, 24 files:	**837**	**27.6**

CSS	SLOC	KB
about-page.css	41	0.5
album.css	161	2.3
artist.css	120	1.9
basic.css	34	0.5
bootstrap-mods.css	55	0.8
edit-mode.css	15	0.2
font-imports.css	53	1.3
form.css	50	0.6
header.css	4	0.1
modal.css	7	0.1
sticky-footer.css	9	0.1
style.css	19	0.4
user-menu.css	41	0.5
variables.css	3	0.0
SUM, 15 files:	**612**	**9.3**

JavaScript	SLOC	KB
displayPage.js	71	2.4
eventsForAlbumForm.js	65	2.4
eventsForArtistForm.js	42	1.6
eventsForUserForm.js	43	1.6
main.js	46	1.6
render404.js	14	0.4
renderAboutFromReadMe.js	24	0.7
renderAlbumForm.js	70	2.7
renderAlbums.js	103	3.9
renderArtistForm.js	40	1.5
renderArtists.js	96	3.5
renderLoginForm.js	53	1.9
renderNavbar.js	60	2.3
renderSpotifyPlaylist.js	35	1.2
renderUserForm.js	55	1.9
renderUserMenu.js	52	1.7
utils/addEventListener.js	14	0.7
utils/addRelations.js	16	0.6
utils/closeHamburgerBar.js	8	0.2
utils/displayToast.js	24	0.9
utils/fetchHelpers.js	49	1.3
utils/fileToBase64.js	11	0.4
utils/followExternalLinksInNewTab.js	6	0.2
utils/formDataCollector.js	22	0.7
utils/navigate.js	8	0.2
utils/noCommasOnArrayToString.js	2	0.1
utils/setActiveMenuChoice.js	13	0.6
utils/usePushStateOnInternalLinks.js	15	0.5
utils/waitForModalAnswer.js	43	1.6
SUM, 30 files:	**1,100**	**39.3**

```
----------------------------------------
|    TOTAL STATISTICS                  |
|       69 files                       |
|       76 kilobytes of source code    |
|    2,549 source lines of code        |
----------------------------------------
```