Dit svar IKKE AFLEVERET

Afleveringsopgave

The Second Mandatory Assignment constitutes the Exam Project. It is individual.

Assignment: create a REST API in PHP8 for part of the Chinook database, which represents a digital media store. Chinook is available for 7 different relational database management systems. Although we have used MySQL throughout the course, feel free to choose the RDBMS you feel more comfortable with. The “AutoIncrementPKs” version of the creation and population script are recommended.

Only the album/track/playlist part of the database will be addressed. The following endpoints must be implemented:

Method	Endpoint	POST/PUT parameters	Further information
GET	artists ✓		Retrieves all artists
GET	artists?s=<search_text> ✓		Retrieves artists whose name includes the search text
GET	artists/<artist_id> ✓		Retrieves one artist
GET	artists/<artist_id>/albums		Retrieves all albums by an artist
POST	artists ✓	name	Creates an artist
DELETE	artists/<artist_id> ✓		Deletes an artist, only if it does not have albums
GET	albums		Retrieves all albums, including their artists
GET	albums?s=<search_text>		Retrieves albums whose title includes the search text, including their artists
GET	albums/<album_id>		Retrieves one album, including its artist
GET	albums/<album_id>/tracks		Retrieves all tracks in an album, including their media types and genres
POST	albums	title, artist_id	Creates an album
PUT	albums/<album_id>	title?, artist_id?	Edits either the title or the artist of an album
DELETE	albums/<album_id>		Deletes an album, only if it does not have tracks
GET	tracks?s=<search_text>		Retrieves tracks whose name includes the search text, including their media types and genres
GET	tracks/<track_id>		Retrieves one track, including its media type and genre
GET	tracks?composer=<composer>		Retrieves tracks by a specific composer
POST	tracks	name, album_id, media_type_id, genre_id, composer, milliseconds, bytes, unit_price	Creates a track
PUT	tracks/<track_id>	name?, album_id?, media_type_id?, genre_id?, composer?, milliseconds?, bytes?, unit_price?	Edits track information
DELETE	tracks/<track_id>		Deletes a track, only if it does not belong to a playlist
GET	media_types		Retrieves all media types

Method	Endpoint	POST/PUT parameters	Further information
GET	genres		Retrives all genres
GET	playlists		Retrieves all playlists
GET	playlists?s=<search_text>		Retrieves playlists whose name includes the search text
GET	playlists/<playlist_id>		Retrieves one playlist, including its tracks
POST	playlists	name	Creates a playlist
POST	playlists/<playlist_id>/tracks	track_id	Assigns a track to a playlist
DELETE	playlists/<playlist_id>/tracks/<track_id>		Removes a track from a playlist
DELETE	playlists/<playlist_id>		Deletes a playlist, only if it does not contain tracks

Because of the security issues derived from reading PUT parameters from the input stream (`file_get_contents("php://input")`), feel free to implement PUT parameters as POST.

Requirements:

- An object-oriented approach will be favoured. Communication with the database will take place in classes.
- Proper error and exception management and the return of proper HTTP codes and error messages is expected.
- The usual security attacks (SQL-injection, XSS) will be properly handled.
- The application will include a logging functionality that will log all requests to file.
- The API should be deployable anywhere in an Apache web server's document root folder structure.
- The API must be deployed on the internet.

Deadline: Sunday 1 June, 23:59 h.