

Katalogi

android

Katalog projektu aplikacji na androida. Testowana na urządzeniach `1080x1920`, na innych rozdzielczościach może się nieprawidłowo wyświetlać.

backend

Katalog zawierający skrypty `PHP` odpowiedzialne za:

- funkcje autoryzacyjne: `auth.php` umożliwia autoryzację jednego z predefiniowanych użytkowników (`wsb` / `user` / `admin`) na podstawie HTTP Basic Auth. Hasło użytkownika jest takie samo jak login
- logowanie: `login.php` umożliwia sprawdzenie czy podane dane logowania są prawidłowe
- zapis wyniku: `save.php` dopisuje nowy rekord do listy wyników w pliku `scores.json`
- pobieranie stanu gry: `index.php` generujący losowe wyrażenie arytmetyczne (operacje na liczbach całkowitych, możliwe operatory: `+-*%`), wynik wyrażenia oraz 3 dodatkowe, losowe wyniki w zakresie +/- 20% od prawidłowego wyniku (zawsze różne niż prawidłowy)

Oraz dodatkowo:

- `composer.json` z zależnościami composera dla `index.php`
- `scores.json` w którym są zapisywane wyniki gier (login, czas gry i

data)

Działanie/założenia

- lista wyników jest dostępna niezależnie od zalogowania
 - pozostałe akcje: pobranie gry i zapis wyniku możliwe są tylko po uprzedniej autoryzacji
 - wynik (czas) gry jest zapisywany automatycznie w momencie pozytywnego zakończenia gry
 - kod domenowy: przetwarzanie stanu gry (`GameService`), autoryzacja (`Auth`) i `SQLiteHelper` są odseparowane od warstwy prezentacji
 - przed zalogowaniem istnieje możliwość wyboru trybu gry: lokalnie (wyniki w bazie, proste wyrażenia) lub przez internet (wyniki na serwerze, skomplikowane wyrażenia generowane po stronie serwera)
 - konfiguracja serwisów domenowych przechowywana jest w uproszczonym kontenerze dependency injection
 - aplikacja na androida napisana w `Kotlinie`, strona serwerowa w `PHP`
- 7

Tryb online

- grać można tylko po prawidłowym zalogowaniu jako jeden z predefiniowanych użytkowników
- komunikacja sieciowa odbywa się przez REST API:
 - żądania są wysyłane jako `application/x-www-form-urlencoded`

- odpowiedzi są wysyłane jako `application/json`
- w przypadku braku autoryzacji/błędnych danych zwracany jest odpowiedni kod HTTP (`401` / `403`)
- dane dynamiczne (stan gry, wyniki i użytkownicy) są przetwarzani po stronie serwera dla większego bezpieczeństwa i spójności

Tryb offline

- Lista wyników przechowywana w lokalnej bazie SQLite
- nie wymaga dostępu do internetu
- logowanie możliwe przez dowolny serwis autoryzacyjny spośród dostępnych
 - `ApiAuth` - autoryzacja via API (to samo co w trybie online)
 - `DummyAuth` - autoryzacja na podstawie danych zapisanych “na sztywno” w kodzie
 - `SQLiteAuth` - autoryzacja na podstawie bazy danych
- domyślny serwis autoryzacyjny to `SQLiteAuth`, aby go zmienić wystarczy podmienić fabrykę/instancję w definicji w `MainApplication`