

# LibraryAPI 2.0

Twoim zadaniem będzie dostosowanie się do nowego interfejsu dostępowego miejskiej biblioteki.

W kodzie źródłowym (github - branch [adapter-task-start](#)) dla tego zadania znajdują się dwa interfejsy: `BookConnector` i `LibraryAPI`.

`BookConnector` to nasz interfejs, natomiast `LibraryAPI` jest dotychczasowym interfejsem dostępowym do biblioteki miejskiej. Nadszedł w niej jednak czas zmian, a wraz z nim pojawiła się druga wersja interfejsu dostępowego - `LibraryAPIv2`.

Należy utworzyć adapter, który przekształci `BookConnector` tak, aby mógł współpracować z nowym interfejsem `LibraryAPIv2`.

Istniejący `BookConnector` operuje na bazie `User` oraz `LibraryAPI`, np.:

- `LibraryAPI api = new LibraryAPIImpl();`
- `User user = new User("Paweł", "Cwik", "32131212");`
- `BookConnector connector = new BookConnector(user, api);`
- `connector.checkAvailability("Harry Potter i Zakon Feniksa");`

gdzie przykładowa implementacja wygląda następująco:

- `public class BookConnector {`
- `private final User user;`
- `private final LibraryAPI api;`

- `public BookConnector(User user, LibraryAPI api) {`
- `this.user = user;`
- `this.api = api;`
- `}`
- `public boolean checkAvailability(String title) {`
- `return api.isAviable(title);`
- `}`
- `}`
- `public interface LibraryAPI {`
- `boolean isAviable(String bookTitle);`
- `LocalDate dueDate(String bookTitle, String pesel);`
- `boolean reserve(String bookTitle, String pesel);`
- `}`

Twoim zdaniem jest utworzenie Adaptera tak, by możliwe były następujące wywołania w metodzie `main`:

- `User user = new User("Paweł", "Cwik", "32131212");`
- `LibraryAPIv2 libraryAPIv2 = new LibraryAPIv2Impl();`
- `APIAdapter apiAdapter = new APIAdapter(libraryAPIv2, user);`
- `BookConnector connector = new BookConnector(user, apiAdapter);`
- `connector.checkAvailability("Harry Potter i Zakon Feniksa");`

Zmiany dopuszczalne są w klasach `User` oraz oczywiście `APIAdapter`, która musi zostać utworzona.

Klasa `BookConnector` ma pozostać nienaruszona.