

Проект: "Словарный запас" (Besediček)

Цель проекта: Разработать базу данных для приложения, которое позволяет изучать новые слова, которые можно будет сохранять в соответствии с выбранными для изучения языками. Создавать в для пользователя карточки слов для изучения и повторения, архивировать выученные слова и работать с несколькими языками сразу.

Часть 1: Проектирование схемы базы данных

1. Модель "Users"

Название таблицы в БД (users)

Поля:

- id (Уникальный идентификатор пользователя, целочисленный, первичный ключ, автоинкрементируемый)
- email (Электронная почта пользователя, тип данных строковый, значение уникальное, поле обязательное к заполнению)
- password (Хэшированный пароль пользователя, тип данных строковый, поле обязательное к заполнению)
- created_at (Дата и время создания записи в таблице в формате UTC, временной штамп: YYYY-mm-ddTHH:MM:SS.mmmZ)
- updated_at (Дата и время обновления записи в таблице в формате UTC, временной штамп: YYYY-mm-ddTHH:MM:SS.mmmZ)

Связи:

- Один ко многим с моделями "User Cards", " User Vocabulary", "User Languages" (один пользователь может быть ответственным за много карточек, словарных запасов, языков)
- Многие ко многим с моделью "Languages" (связь между таблицей users и таблицей languages осуществляется через таблицу user_languages)

2. Модель "Languages"

Название таблицы в БД (languages)

Поля:

- code (Первичный ключ, 3х буквенный языковой код по ISO 639-3, тип данных строковый, значение уникальное, обязательное к заполнению поле)
- name (Поле для полного названия языка, тип данных строковый, значение уникальное, обязательное к заполнению поле)
- created_at (Дата и время создания записи в таблице в формате UTC, временной штамп: YYYY-mm-ddTHH:MM:SS.mmmZ)
- updated_at (Дата и время создания записи в таблице в формате UTC, временной штамп: YYYY-mm-ddTHH:MM:SS.mmmZ)

Связи:

- Один ко многим с моделями "User Languages" (один пользователь может иметь много выбранных языков)
- Многие ко многим с моделью "Users" (связь между таблицей users и таблицей languages осуществляется через таблицу user_languages)

3. Модель "User Languages"

Название таблицы в БД (user_languages)

Поля:

- id (Уникальный идентификатор языка пользователя, целочисленный, первичный ключ, автоинкрементируемый)
- user_id (Внешний ключ, ссылается на id в таблице users)
- lang_code (Внешний ключ, ссылается на code в таблице languages)
- native_lang (Логическое поле, указывающее, является ли язык родным для пользователя, булев тип данных, обязательное для заполнения поле)
- active_lang (Логическое поле, указывающее, активен ли язык, булев тип данных, обязательное для заполнения поле)
- created_at (Дата и время создания записи в таблице в формате UTC, временной штамп: YYYY-mm-ddTHH:MM:SS.mmmZ)
- updated_at (Дата и время создания записи в таблице в формате UTC, временной штамп: YYYY-mm-ddTHH:MM:SS.mmmZ)

Связи:

Один ко многим с моделями "User Cards", "User Vocabulary", "Users", "Languages" (один пользователь может быть ответственным за много карточек, словарных запасов, языков)

4. Модель "User Vocabulary"

Название таблицы в БД (user_vocabulary)

Поля:

- id (Уникальный идентификатор записи в словаре пользователя, целочисленный, первичный ключ, автоинкрементируемый)
- lang_code (Внешний ключ, связанный с lang_code таблицы user_languages)
- word (Строковое поле для слова, обязательное для заполнения)
- note (Текстовое поле для добавления комментария по слову: перевода, примера, описания)
- created_at (Дата и время создания записи в таблице в формате UTC, временной штамп: YYYY-mm-ddTHH:MM:SS.mmmZ)
- updated_at (Дата и время обновления записи в таблице в формате UTC, временной штамп: YYYY-mm-ddTHH:MM:SS.mmmZ)

Связи:

Много к одному с моделью "User_languages" (связь через поле lang_code). Много записей в таблице "User Vocabulary" могут ссылаться на одну запись в таблице "User_languages".

5. Модель "User Cards"

Название таблицы в БД (user_cards)

Поля:

- id (Уникальный идентификатор карты пользователя, целочисленный, первичный ключ, автоинкрементируемый)
- user_id (Внешний ключ, связанный с id таблицы users)
- title (Строковое поле для названия карты, обязательное для заполнения)
- status (Статус карточки, строковое поле, может принимать одно из трех значений: 'NEW', 'IN

PROGRESS', 'DONE')

- created_at (Дата и время создания записи в таблице в формате UTC, временной штамп: YYYY-mm-ddTHH:MM:SS.mmmZ)
- updated_at (Дата и время обновления записи в таблице в формате UTC, временной штамп: YYYY-mm-ddTHH:MM:SS.mmmZ)

Связи:

Много к одному с моделью "Users" (связь через поле user_id)

6. Модель "User Word Cards"

Название таблицы в БД (user_word_cards)

Поля:

- id (Уникальный идентификатор слова в карточке пользователя, целочисленный, первичный ключ, автоинкрементируемый)
- user_vocabulary_id (Внешний ключ, связанный с id таблицы user_vocabulary)
- card_id (Внешний ключ, связанный с id таблицы user_cards)
- word_status (Статус слова в карточке, строковое поле, может принимать одно из трех значений: 'NEW', 'IN PROGRESS', 'DONE')
- created_at (Дата и время создания записи в таблице в формате UTC, временной штамп: YYYY-mm-ddTHH:MM:SS.mmmZ)
- updated_at (Дата и время обновления записи в таблице в формате UTC, временной штамп: YYYY-mm-ddTHH:MM:SS.mmmZ)

Связи:

Много к одному с моделью "User Cards", "User Vocabulary" (связь через поле card_id и поле user_vocabulary_id соответственно)

Часть 2: Функциональность проекта

Приложение "Besediček" обеспечит пользователям эффективный способ изучения новых слов на разных языках и отслеживания их прогресса в изучении. Необходимо разработать следующий список функциональности для данного приложения:

1. Авторизация и аутентификация пользователя:

Пользователь должен иметь возможность зарегистрироваться или войти в систему с помощью своего email и пароля.

2. Пользователи могут выбирать свой родной язык из предложенных вариантов.

На выбор следующие языки:

Русский, Белорусский, Сербский, Словенский, Итальянский, Английский, Немецкий.

3. Выбор языка для нового слова:

Пользователи могут выбирать языки для личного словаря, в которые они хотят добавлять новые

слова. При добавлении языка в свой кабинет, язык приобретает статус "активный" (в БД), и иконка (код) языка отображается на дашборде (на домашней странице пользователя). При нажатии на иконку пользователь переходит на страницу пользовательского словаря этого языка.

Пользователи могут деактивировать и снова активировать языки, которые они изучают. При деактивации значок языка исчезает с дашборда и может быть снова активирован только при переходе на страницу "Языки пользователя".

4. Добавление, редактирование и удаление слов в/из словаря выбранного языка:

Пользователи могут добавлять новые слова в свой словарь на выбранном языке.

Есть возможность просмотра всего словаря, редактирования и удаления добавленных слов.

5. Создание карточек и их статус:

Пользователи могут создавать карточки слов для изучения, в которые они могут добавлять слова.

Карточка должна иметь уникальное название, и может содержать слова из разных словарей (языков). При создании карточки она приобретает статус NEW. При изменении статуса любого слова в карточке отправляется запрос с проверкой: если все слова в карточке находятся в статусе NEW, то карточка должна иметь статус NEW; если все слова в карточке находятся в статусе DONE, то карточка переводится в статус DONE; иначе карточка имеет статус IN PROGRESS.

6. Просмотр всех карточек пользователем:

Созданные карточки представлены на дашборде пользователя, отсортированные в порядке убывания по дате обновления. Если пользователь имеет больше 9 карточек (размещение по 3 карты в ряд) для изучения, то на странице появляется пагинация.

7. Обновление и удаление карточек:

Пользователь может переименовывать карточки и удалять их. При удалении карточки записи удаляются из обеих таблиц: `user_cards` (удаление строки по `id` карточки) и `user_word_cards` (все строки, привязанные к `card_id`).

8. Добавление слов в карточку для изучения:

Пользователи могут добавлять в карточку слова из своих словарей. Максимальное количество добавленных слов в карточку равно 10. Слово добавленное в карточку получает статус NEW.

9. Просмотр и редактирование слов в карточке:

Пользователи могут просматривать слова в созданных ими карточках.

Пользователи могут менять статус слов в карточках с NEW на IN PROGRESS или DONE.

10. Удаление слов из карточки для изучения:

Пользователи могут удалять слова из карточек для изучения.

Часть 3: Описание эндпоинтов

1. Эндпоинт: `/api/auth/login`

- Метод: POST
- Описание: Аутентификация пользователя.
- Входные данные:
email (строка, обязательное поле)
password (строка, обязательное поле)
- Преобразование данных: Проверка соответствия введенных данных с данными в базе данных.
- Запись в таблицы: Нет.
- Валидация данных: Проверка формата электронной почты и наличия пароля.

После логина пользователь получает **access_token**, который содержит информацию о текущем пользователе, включая **user_id**. Данный токен имеет срок действия и должен передаваться при всех запросах пользователя в заголовке запроса (Authorization: Bearer {access_token})

2. Эндпоинт: /api/auth/register

- Метод: POST
- Описание: Регистрация нового пользователя.
- Входные данные:
email (строка, обязательное поле)
password (строка, обязательное поле)
- Преобразование данных: Создание новой записи в таблице users с указанными данными.
- Запись в таблицы: Таблица users.
- Валидация данных: Проверка формата электронной почты, уникальности email и длины пароля.

3. Эндпоинт: /api/languages

- Метод: GET
- Описание: Получение списка всех доступных языков приложения.
- Входные данные: Нет.
- Преобразование данных: Получение списка языков из таблицы languages
- Запись в таблицы: Нет.
- Валидация данных: Не требуется.

4. Эндпоинт: `/api/users/languages`

- Методы:

GET: Получить список языков пользователя.

POST: Создать новую запись о языке пользователя.

- Описание: Получение списка языков пользователя или добавление новой записи о языке для указанного пользователя.

- Входные данные (POST):

`lang_code` (3х буквенный код, обязательное поле): Буквенный код языка.

`native_lang` (логическое значение, обязательное поле): Указывает, является ли язык родным для пользователя.

`active_lang` (логическое значение, обязательное поле): Указывает, активен ли язык для пользователя.

- Преобразование данных (POST): Создание новой записи в таблице `user_languages` с указанными данными.

- Запись в таблицы (POST): Таблица `user_languages`.

- Валидация данных (POST): Проверка существования пользователя, языка и корректности логических значений.

- Выходные данные (GET): Список языков пользователя с дополнительной информацией о каждом языке.

Пример запроса:

POST `/api/users/languages` HTTP/1.1

Host: {host_name}

Authorization: Bearer {user_access_token}

Content-Type: application/json

```
{
  "lang_code": "ENG",
  "native_lang": true,
  "active_lang": true
}
```

user_id должен быть передан в качестве части запроса через аутентификацию в заголовке запроса во всех запросах пользователя, где `user_access_token` представляет собой токен доступа, который содержит информацию о текущем пользователе, включая `user_id`.

5. Эндпоинт: `/api/users/languages/{lang_code}`

- Методы:

GET: Получить информацию о языке пользователя.

PUT: Обновить информацию о языке пользователя.

- Описание: Получение информации о языке пользователя и обновление записи.

- Входные данные (PUT):

`lang_code` (3х буквенный код, обязательное поле): Код языка пользователя.

`native_lang` (логическое значение): Указывает, является ли язык родным для пользователя.

`active_lang` (логическое значение): Указывает, активен ли язык для пользователя.

- Преобразование данных (PUT): Обновление записи в таблице `user_languages` с указанными данными.

- Валидация данных (PUT): Проверка существования пользователя, языка и корректности логических значений.

6. Эндпоинт: `/api/users/languages/{lang_code}/vocabulary`

- Методы:

GET: Получить список всех слов в словаре пользователя для определенного языка с возможностью пагинации.

POST: Добавить новое слово в словарь пользователя для определенного языка.

- Описание: Получение списка всех слов в словаре пользователя для определенного языка с возможностью пагинации, добавление нового слова.

- Входные данные (POST):

`lang_code` (3-буквенный код, обязательное поле): Код языка словаря пользователя.

`word` (строка, обязательное поле): Слово для добавления в словарь.

`note` (строка, необязательное поле): Дополнительная информация о слове (перевод, пример, комментарий).

- Запись в таблицы (POST): Таблица `user_vocabulary`. Добавление новой записи в таблицу `user_vocabulary` для текущего пользователя и указанного языка.

- Валидация данных (POST): Проверка наличия обязательных полей: `lang_code`, `word`.

- Выходные данные (GET): Список всех слов в словаре пользователя для определенного языка с учетом примененной пагинации

Параметры пагинации:

`current_page` (число, необязательное поле): Номер страницы для отображения.

`total_pages` (число, необязательное поле): Количество существующих страниц.

`items_per_page` (число, необязательное поле): Максимальное количество элементов на странице.

`total_items` (число, необязательное поле): Количество существующих элементов в целом.

Пример пагинации:

```
"pagination": {"total_pages": "", "current_page": "", "total_items": "", "items_per_page": 10}
```

7. Эндпоинт: `/api/users/ languages/{lang_code}/vocabulary/{word_id}`

- Методы:

GET: Получить информацию о слове в словаре пользователя для определенного языка.

PUT: Обновить информацию о слове в словаре пользователя для определенного языка.

DELETE: Удалить слово из словаря пользователя для определенного языка.

- Описание: Получение информации о конкретном слове в словаре пользователя выбранного языка, обновление информации данного слова или удаление слова из словаря.

- Входные данные (PUT):

lang_code (3-буквенный код, обязательное поле): Код языка пользователя передается в URL запроса.

word_id (идентификатор слова, обязательное поле): Уникальный идентификатор слова в словаре пользователя.

word (строка): Текстовое представление слова.

note (строка): Перевод слова на язык пользователя или комментарий к слову.

- Преобразование данных (PUT): Обновление записи в таблице user_vocabulary с указанными данными.

- Валидация данных (PUT): Проверка существования пользователя, языка, слова и корректности логических значений.

- Входные данные (DELETE):

lang_code (3-буквенный код, обязательное поле): Код языка пользователя передается в URL запроса.

word_id (идентификатор слова, обязательное поле): Уникальный идентификатор слова в словаре пользователя.

- Преобразование данных (DELETE): Удаление записи из таблицы user_vocabulary соответствующего указанным данным.

- Валидация данных (DELETE): Проверка существования пользователя, языка и слова перед удалением.

8. Эндпоинт: `api/users/cards`

- Методы:

GET: Получить все карточки пользователя с возможностью пагинации.

POST: Создать новую карточку пользователя.

- Описание: Получать информацию о всех карточках пользователя и добавлять новую карточку для пользователя с указанным идентификатором.
- Входные данные (POST):
card_title (строка, обязательное поле): Название новой карточки.
- Запись в таблицы (POST): Создание новой записи в таблице user_cards с указанным идентификатором пользователя и названием карточки.
- Валидация данных (POST): Проверка существования указанного пользователя. Проверка наличия обязательных полей: card_title.
- Выходные данные (GET): Список всех карточек пользователя с учетом примененной пагинации
Параметры пагинации:
current_page (число, необязательное поле): Номер страницы для отображения.
total_pages (число, необязательное поле): Количество существующих страниц.
items_per_page (число, необязательное поле): Максимальное количество элементов на странице.
total_items (число, необязательное поле): Количество существующих элементов в целом.

Пример пагинации:

```
"pagination": {"total_pages": "", "current_page": "", "total_items": "", "items_per_page": 10}
```

9. Эндпоинт api/users/cards/{card_id}

- Методы:

GET: Получить информацию о карточке пользователя.

PUT: Обновить информацию о карточке пользователя.

DELETE: Удалить карточку пользователя.

- Описание:

Предоставляет возможность получения информации о карточке слов пользователя, изменения названия и статуса карточки или ее удаления из профиля пользователя.

- Входные данные (PUT):

card_id (идентификатор карточки, обязательное поле): Уникальный идентификатор карточки пользователя.

title (строка): Название карточки пользователя.

status (Статус карточки, строковое поле, может принимать одно из трех значений: 'NEW', 'IN PROGRESS', 'DONE')

- Преобразование данных (PUT):

Обновление записи в таблице user_cards с указанными данными.

- Валидация данных (PUT):

Проверка существования пользователя и корректности данных карточки.

- Входные данные (DELETE):

card_id (идентификатор карточки, обязательное поле): Уникальный идентификатор карточки пользователя.

- Валидация данных (DELETE):

Проверка существования card_id в таблице user_word_cards перед удалением.

- Преобразование данных (DELETE):

Удаление записи из таблицы user_cards с указанным card_id и из таблицы user_word_cards с соответствующими card_id, если представлены

10. Эндпоинт: `api/users/cards/{card_id}/words`

- Методы:

GET: Получить все слова на карточке пользователя.

POST: Добавить новое слово на карточку пользователя.

- Описание:

GET: Получить информацию о всех словах на карточке пользователя с их идентификаторами, статусами и другими деталями.

POST: Добавить новое слово на карточку пользователя с указанным идентификатором.

- Входные данные (POST):

user_vocabulary_id (число, обязательное поле): Идентификатор пользовательского словаря, из которого будет добавлено слово

card_id (число, обязательное поле): Идентификатор карточки, в которую добавляется слово.

word_status (строка, обязательное поле): Статус слова в карточке. Может принимать одно из трех значений: 'new', 'learning', 'archived'.

- Запись в таблицы (POST): Создание новой записи в таблице user_word_cards с указанными данными.

- Валидация данных (POST): Проверка существования указанного пользователя и карточки. Проверка наличия обязательных полей: user_vocabulary_id, word_status.

11. Эндпоинт: `api/users/cards/{card_id}/words/{word_id}`

- Методы:

GET: Получить информацию о слове в карте пользователя.

PUT: Обновить статус слова в карте пользователя.

DELETE: Удалить слово из карты пользователя.

- Описание:

Получение информации о конкретном слове в карте пользователя, обновления статуса данного слова или удаления слова из карты.

- Входные данные (PUT):

card_id (идентификатор карточки, обязательное поле): Уникальный идентификатор карточки пользователя.

word_id (идентификатор слова, обязательное поле): Уникальный идентификатор слова в карте пользователя.

word_status (строка): Новый статус слова в карте пользователя. Может принимать одно из трех значений: 'NEW', 'IN PROGRESS', 'DONE'.

- Преобразование данных (PUT):

Обновление статуса в таблице user_word_cards с указанными данными.

- Валидация данных (PUT):

Проверка существования слова в карточке.

- Входные данные (DELETE):

card_id (идентификатор карточки, обязательное поле): Уникальный идентификатор карточки пользователя.

word_id (идентификатор слова, обязательное поле): Уникальный идентификатор слова в карте пользователя.

- Преобразование данных (DELETE):

Удаление записи из таблицы user_word_cards соответствующей указанным данным.

- Валидация данных (DELETE):

Проверка существования слова перед удалением.