1. Поясните понятие «ORM»

технология программирования, которая позволяет работать с SQL-базой данных, как с набором программных объектов. Mapping: база данных – объект контекста, таблица – коллекция объектов, строка в таблице – объект, структура таблицы – класс

1. Поясните понятие «модель».

это абстракция, представляющая таблицу в базе данных. Модель сообщает несколько вещей о сущности, которую она представляет, например, имя таблицы в базе данных и какие столбцы у нее есть (и их типы данных). У модели есть имя. Это имя не обязательно должно совпадать с именем таблицы, которую она представляет в базе данных. Обычно модели имеют имена в единственном числе (например, User), а таблицы – имена во множественном числе (например, Users)

1. Поясните понятие «пул соединений» и его назначение.
2. Что такое Sequelize и Prisma?

Sequelize- это ORM-библиотека, основанная на промисах, для приложений на Node.js, которая осуществляет сопоставление таблиц в БД и отношений между ними с классами. Может применяться для: Postgres, MySQL, MariaDB, SQLite, Microsoft SQL Server, Oracle, DB2, Snowflake. Поддерживает TS и JS.

1. Поясните, что такое «scope» в Sequelize?

Области видимости, скоупы используются, чтобы помочь повторно использовать код. Можно определить часто используемые запросы, указав такие параметры, как where, include, limit и т.д. Их можно применить к findAll, count, update, destroy,…. Скоупы бывают обычными и по умолчанию. Скоупы определяются при создании модели (options.scopes, options.defaultScope) или с помощью Model.addScope(). Могут быть объектами поиска или функциями, возвращающими объекты поиска, за исключением скоупа по умолчанию (options.defaultScope), который может быть только объектом.

1. Поясните, что такое «хуки» в Sequelize? Какие они бывают?

Хуки, или события жизненного цикла представляют собой функции, которые выполняются до/после/во время действий с данными. Хуки бывают глобальные постоянные (для всех моделей), глобальные по умолчанию (используется для модели если она не определяет свой собственный хук), локальные хуки инстанса и локальные хуки модели. Локальные хуки всегда запускаются перед глобальными хуками. Определение хуков модели осуществляется при ее создании (options.hooks) либо с помощью Model.addHook() (удаление через Model.removeHook()).

1. Поясните назначение функции Sequelize *Модель*.**hasMany**.

Sequelize поддерживает стандартные ассоциации: One-To One, One To-Many и Many-To-Many.

Для этого Sequelize предоставляет четыре типа ассоциаций, которые

следует комбинировать для их создания:

• HasOne

• BelongsTo

• HasMany

• BelongsToMany

Изображение выглядит как текст, снимок экрана, Шрифт, программное обеспечение

Автоматически созданное описание

1. Поясните понятие «raw query» в Sequelize.

Изображение выглядит как текст, снимок экрана, Шрифт, число

Автоматически созданное описание

1. Назовите отличия Prisma от других фреймворков ORM?

это ORM-библиотека, нового поколения с открытым исходным кодом для Node.js и TypeScript. Она состоит из следующих инструментов: • Prisma Client: автогенерируемый и типобезопасный клиент базы данных; • Prisma Migrate: система миграций; • Prisma Studio: пользовательский интерфейс для = просмотра и редактирования данных.

Изображение выглядит как текст, снимок экрана, Шрифт

Автоматически созданное описание

1. Можно ли использовать Prisma с базами данных SQL и NoSQL?

Prisma можно использовать только с базами данных SQL, такими как PostgreSQL, MySQL, SQLite и MSSQL. Prisma не поддерживает работу с NoSQL базами данных, такими как MongoDB или Cassandra.

1. Какие языки программирования совместимы с Prisma?

TypeScript, JavaScript;

1. Расскажите, где и как можно использовать ***every***, ***none*** и ***some*** вPrisma?

Изображение выглядит как текст, Шрифт, снимок экрана

Автоматически созданное описание

Изображение выглядит как текст, Шрифт, снимок экрана, линия

Автоматически созданное описание

Изображение выглядит как текст, Шрифт, снимок экрана

Автоматически созданное описание

Изображение выглядит как текст, Шрифт, снимок экрана, линия

Автоматически созданное описание



1. Для чего нужен метод ***upsert*** вPrisma?

Метод upsert в Prisma используется для обновления записи в базе данных, если она уже существует, или создания новой записи, если ее еще нет. Таким образом, данный метод позволяет сократить количество кода, необходимого для выполнения операций обновления и создания записей в базе данных. Upsert также может быть полезен при работе с уникальными полями, такими как email или username, когда нужно обновлять данные пользователя или создавать нового пользователя, если его email или username уже существуют в базе данных.

1. Как включить регистронезависимый поиск вPrisma?

Для включения регистронезависимого поиска в Prisma необходимо использовать функцию "toLowerCase()" для приведения всех символов в строке к нижнему регистру. Например, для поиска записей в таблице "Users" по имени пользователя, можно использовать следующий код:

const users = await prisma.users.findMany({

where: {

username: {

contains: searchTerm.toLowerCase()

}

}

})

В данном примере, переменная "searchTerm" содержит строку поискового запроса. Функция "toLowerCase()" применяется к этой строке, чтобы привести ее к нижнему регистру. Затем, используется оператор "contains" для поиска записей, содержащих строку "searchTerm" в имени пользователя, без учета регистра символов.

1. Объясните назначение методов ***$queryRaw***, ***$executeRaw*** вPrisma.

Метод $queryRaw в Prisma используется для выполнения произвольных SQL-запросов к базе данных, которые не могут быть выполнены с помощью стандартных методов Prisma. Этот метод позволяет передавать произвольный SQL-запрос в виде строки и получать результаты в виде объектов, которые можно обрабатывать в коде.

Метод $executeRaw также используется для выполнения произвольных SQL-запросов, но в отличие от метода $queryRaw он не возвращает результаты запроса. Вместо этого, этот метод возвращает количество затронутых строк в базе данных.

Оба метода могут быть полезны при работе с базой данных, когда необходимо выполнить сложный запрос или выполнить операции, которые не могут быть выполнены с помощью стандартных методов Prisma. Однако, использование этих методов может привести к потере некоторых преимуществ, которые предоставляет Prisma, таких как безопасность и проверка типов данных.

1. Что такое аутентификация?

процедура проверки подлинности, то есть доказательство того, что пользователь именно тот, за кого себя выдает. Чтобы определить чью-то подлинность, можно воспользоваться тремя факторами: • Пароль – что-то, известное только пользователю (слово, PIN-код, код для замка, = графический ключ) • Устройство – что-то, имеющееся только у пользователя (токен, пластиковая карта, ключ от замка) • Биометрика – что-то, присущее только пользователю (отпечаток пальца, сетчатка глаза, сканер лица)

1. Что такое токен? Какие бывают?

битовая последовательность, построенная по определенному принципу, которая позволяет точно идентифицировать объект и определить уровень его привилегий. Токен обычно содержит в себе следующую информацию Issuer (кто выдал), Audience (кому выдан), Expires On (время жизни), Claim (сведение о пользователе), подпись (для защиты от изменений и для гарантий подлинности). Применяется, как правило, для реализации Single Sign-On (технология единого входа) в распределенных системах.

Изображение выглядит как текст, Шрифт, снимок экрана, информация

Автоматически созданное описание Изображение выглядит как текст, Шрифт, снимок экрана, информация

Автоматически созданное описание

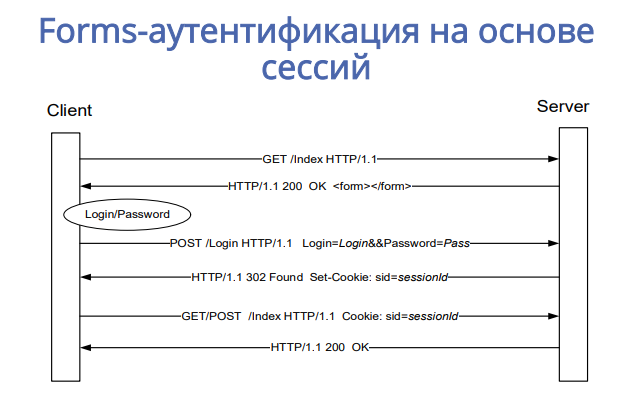
Изображение выглядит как текст, Шрифт, снимок экрана

Автоматически созданное описание

Изображение выглядит как текст, снимок экрана, Шрифт

Автоматически созданное описание

1. Опишите общую схему FORMS-аутентификации на основе сессий.



1. Опишите общую схему FORMS-аутентификации на основе токенов.

Изображение выглядит как текст, снимок экрана, Шрифт, линия

Автоматически созданное описание