

Auth Service

AUTH SERVICE METHOD: TGRegistration
+ http method : POST
+ path : /auth/twa/registration
+ body : null
+ auth header type : twa
+ query params : null
+ description : Регистрация с помощью twa.
+ access : ANY

AUTH SERVICE METHOD: TGRegistration
+ http method : POST
+ path : /auth/tg/registration
+ body : {userName : string}
+ auth header type : servicekey : bot
+ query params : null
+ description : Регистрация через микро сервис бот
+ access : ANY

Telegram Service

Account Service

ACCOUNT SERVICE METHOD: Update

- + http method : PUT
- + path : /account
- + body : {account : Account}
- + auth header type : twa
- + query params : null
- + description : Изменить данные аккаунта
- + access : ANY

Profile Service

PROFILE SERVICE METHOD: create

- + http method : POST
- + path : /profile
- + body : {description : string, text : string, imageUri? : string}
- + auth header type : twa
- + query params : ?id=string
- + description : Создать контент для своего профиля/ портфолио
в query передается id профессии
- + access : MASTER

PROFILE SERVICE METHOD: get

- + http method : GET
- + path : /profile
- + body : null
- + auth header type : twa
- + query params : null
- + description : Показать все профили
- + access : ANY

PROFILE SERVICE METHOD: delete OR adminDelete

- + http method : DELETE
- + path : /profile
- + body : null
- + auth header type : twa
- + query params : ?id=string
- + description : Удалить содержимое контента профиля.
Мастер удаляет только свой. Админ любой.
- + access : MASTER OR ADMIN

PROFILE SERVICE METHOD: update

- + http method : PUT
- + path : /profile
- + body : {profileContent : ProfileContent}
- + auth header type : twa
- + query params : null
- + description : Изменить содержимое контента
профиля
- + access : MASTER



PROFESSION SERVICE METHOD: join
+ http method : POST
+ path : /profession/join
+ body : {}
+ auth header type : twa
+ query params : ?id=string
+ description : Присвоить аккаунту профессию. Роль меняется на MASTER
+ access : ANY

PROFESSION SERVICE METHOD: leave
+ http method : DELETE
+ path : /profession/leave
+ body : null
+ auth header type : twa
+ query params : ?id=string
+ description : Выйти из профессии
+ access : MASTER

Offer Service

OFFER SERVICE METHOD: delete

- + http method : DELETE
- + path : /offer
- + body : null
- + auth header type : twa
- + query params : ?id=string
- + description : Удаление предложения мастера (условно "товара")
- + access : MASTER
- + response : {message : "ok"}

OFFER SERVICE METHOD: get

- + http method : GET
- + path : /offer
- + body : null
- + auth header type : twa
- + query params : null
- + description : Показать все предложения
- + access : ANY
- + response : Offer[]

OFFER SERVICE METHOD: update

- + http method : PUT
- + path : /offer
- + body : {offer : Offer}
- + auth header type : twa
- + query params : null
- + description : Изменение предложения мастера (условно "товара")
- + access : MASTER
- + response : Offer

OFFER SERVICE METHOD: create

- + http method : POST
- + path : /offer
- + body : {professionId : string, price : number, description : string, imageUrl? : string}
- + auth header type : twa
- + query params : null
- + description : Создание предложения мастера (условно "товара")
- + access : MASTER
- + response : offer : Offer

TimeSlotUser Service

TIMESLOTUSER SERVICE METHOD: get

+ http method : GET
+ path : /timeslot
+ body : null
+ auth header type : twa
+ query params : null
+ description : Показать все свободные временные слоты
+ access : ANY
+ response : TimeSlot[]

TIMESLOTUSER SERVICE METHOD: reserve

+ http method : PUT
+ path : /timeslot/reserve
+ body : null
+ auth header type : twa
+ query params : ?id=string
+ description : Забронировать выбранный временной слот
+ access : ANY
+ response : TimeSlotReserve

TIMESLOTUSER SERVICE METHOD: cancel

+ http method : PUT
+ path : /timeslot/cancel
+ body : null
+ auth header type : twa
+ query params : ?id=string
+ description : Снять резервацию со слота
+ access : ANY
+ response : {message : "ok"}

TIMESLOTUSER SERVICE METHOD: my

+ http method : GET
+ path : /timeslot/me
+ body : null
+ auth header type : twa
+ query params : null
+ description : Показать все зарезервированный слоты пользователем
+ access : ANY
+ response : TimeSlotReserve[]

TimeSlotMaster Service

TIMESLOTMASTER SERVICE METHOD: my

+ http method : GET
+ path : /timeslot/master/me
+ body : null
+ auth header type : twa
+ query params : null
+ description : Вернуть все временные слоты мастера
+ access : MASTER
+ response : TimeSlot[]

TIMESLOTMASTER SERVICE METHOD: create

+ http method : POST
+ path : /timeslot/master/
+ body : {offerId : string, startDate : string, endDate : string}
+ auth header type : twa
+ query params : null
+ description : Создать свободный временной слот
+ access : MASTER
+ response : TimeSlot

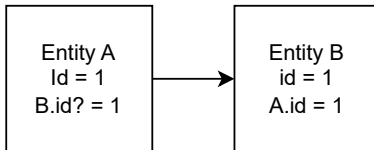
TIMESLOTMASTER SERVICE METHOD: update

+ http method : PUT
+ path : /timeslot/master/
+ body : {timeSlot : TimeSlot}
+ auth header type : twa
+ query params : null
+ description : Изменить свободный временной слот
+ access : MASTER
+ response : TimeSlot

TIMESLOTMASTER SERVICE METHOD: delete

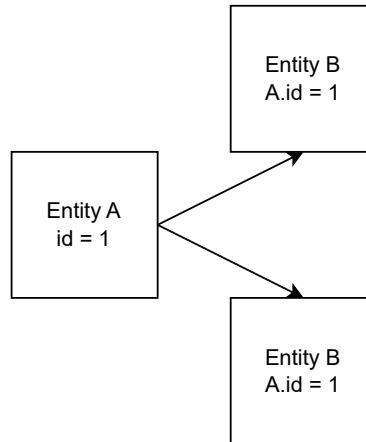
+ http method : DELETE
+ path : /timeslot/master/
+ body : null
+ auth header type : twa
+ query params : ?id=string
+ description : Удалить / закрыть временной слот
+ access : MASTER
+ response : {message : "ok"}

One - To - One Один к одному



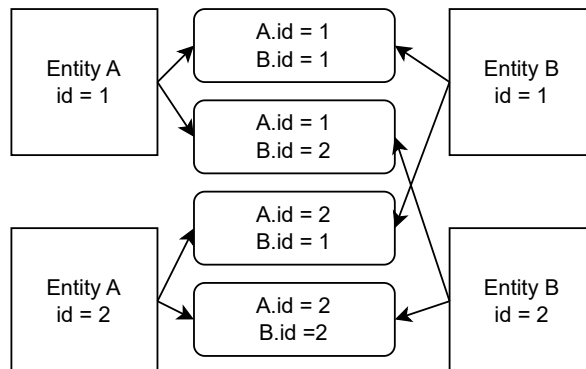
Одному экземпляру А может "принадлежать" один экземпляр В и наоборот
Сущность А имеет одну ссылку на сущность В.
Сущность В может иметь ссылку на ту же сущность А
Обе ссылки должны быть уникальными

Belongs-To-Many Один ко многим



Одному экземпляру А может принадлежать много экземпляров В, А экземпляру В принадлежит только один А
Сущность В имеет ссылку на сущность А
Ссылки не уникальная.
Поэтому разные В могут иметь ссылку на одну и ту же А, поэтому получается что А имеет несколько В

Many-To-Many



Один экз А может иметь много В, но и В может иметь много А
Для реализации используется промежуточная таблица в которой есть ссылки на А и на В

SERVICE METHOD:

```

+ http method :
+ path : /
+ body : {}
+ auth header type : twa
+ query params : ?
+ description :
+ access : ANY
+ response : {message : "ok"}
  
```