<<Interface>> IAccountWebService

- + add(AccountDTO): AccountDTO
- + update(AccountDTO): AccountDTO
- + delete(AccountDTO): void
- + findAll(void): ResponseEntity<List<AccountDTO>>>
- + findById(String): ResponseEntity<AccountDTO>
- + findByUsername(String): ResponseEntity<AccountDTO>
- + findUsers(void): ResponseEntity<List<AccountDTO>
- + findAdmins(void): ResponseEntity<List<AccountDTO>>>
- + findSuperUsers(void): ResponseEntity<List<AccountDTO>>>
- + findAdminsAndSuperUsers(void): ResponseEntity<List<AccountDTO>>>
- + findBanned(void): ResponseEntity<List<AccountDTO>>>
- + findNotBanned(void): ResponseEntity<List<AccountDTO>>>
- + findDeleted(void): ResponseEntity<List<AccountDTO>>
- + findUsersDeleted(void): ResponseEntity<List<AccountDTO>>>
- + findAdminsDeleted(void): ResponseEntity<List<AccountDTO>>
- + findNotDeleted(void): ResponseEntity<List<AccountDTO>>
- + findUsersNotDeleted(void): ResponseEntity<List<AccountDTO>>
- + findAdminsNotDeleted(void): ResponseEntity<List<AccountDTO>>

AccountWebService

accountService: IAccountService

<<Interface>>

IUserWebService

- + add(UserDTO): UserDTO
- + update(UserDTO): UserDTO
- + delete(UserDTO): void
- + findAll(void): ResponseEntity<List<UserDTO>>>
- + findByld(String): ResponseEntity<UserDTO>
- + findByLastName(String): ResponseEntity<List<UserDTO>>
- + findByAccountUsername(String): ResponseEntity<UserDTO>

UserWebService

- userService: IUserService

<<Interface>> IRatingWebService

- + add(RatingDTO): RatingDTO
- + update(RatingDTO): RatingDTO
- + delete(RatingDTO): void
- + findAll(void): ResponseEntity<List<RatingDTO>>
- + findById(String): ResponseEntity<RatingDTO>

RatingWebService

- ratingService: IRatingService

<IVehicleWebService

- + add(VehicleDTO): VehicleDTO
- + update(VehicleDTO): VehicleDTO
- + delete(VehicleDTO): void
- + findAll(void): ResponseEntity<List<VehicleDTO>>
- + findByld(String): ResponseEntity<VehicleDTO>

VehicleWebService

- vehicleService: IVehicleService

dadakar-back-img

<<Interface>> IlmageStorageService

- + storeImage(MultipartFile): void + getImageAsResource(String): ResponseEntity<InputStreamResource> +getImageByFilename(String): ResponseEntity<InputStreamResource> + getAllImages(void): ResponseEntity<List<InputStreamResource>>

ImageStorageService

- imageStorageRepository: IImageStorageRepository

<<Interface>> IRunWebService

- + add(RunDTO): RunDTO
- + update(RunDTO): RunDTO
- + delete(RunDTO): void
- + findAll(void): ResponseEntity<List<RunDTO>>
- + findByld(String): ResponseEntity<RunDTO>
- + findByDriverId(String): ResponseEntity<List<RunDTO>>
- + findNotCancelledByDriverId(String): ResponseEntity<List<RunDTO>>>
- + findByPassengerId(String): ResponseEntity<List<RunDTO>>
- + findNotCancelledByPassengerId(String): ResponseEntity<List<RunDTO>>
- + findByUserId(String): ResponseEntity<List<RunDTO>>>
- + findNotCancelledByUserId(String): ResponseEntity<List<RunDTO>>>
- + findCurrentByUserId(String): ResponseEntity<List<RunDTO>>
- + findCurrentNotCancelledByUserId(String): ResponseEntity<List<RunDTO>>>
- + findPassedBvUserId(String): ResponseEntity<List<RunDTO>>>
- + findPassedNotCancelledByUserId(String): ResponseEntity<List<RunDTO>>>
- + find(String, String, Date, String, String): ResponseEntity<List<RunDTO>>

RunWebService

- runService: IRunService

<<Interface>> IWayPointWebService

- + add(WayPointDTO): WayPointDTO
- + update(WayPointDTO): WayPointDTO
- + delete(WayPointDTO): void
- + findAll(void): ResponseEntity<List<WayPointDTO>>>
- + findById(String): ResponseEntity<WayPointDTO>

WayPointWebService

- wayPointService: IWayPointService

<<interface>> IAddressWebService

- + add(AddressDTO): AddressDTO
- + update(AddressDTO): AddressDTO
- + delete(AddressDTO): void
- + findAll(void): ResponseEntity<List<AddressDTO>>
- + findByld(String): ResponseEntity<AddressDTO>
- + findByTown(String): ResponseEntity<List<AddressDTO>>>
- + findByTownAndDistrict(String, String): ResponseEntity<AddressDTO>

AddressWebService

- addressService: IAddressService

<<Interface>> ITollWebService

- + add(TolIDTO): TolIDTO
- + update(TolIDTO): TolIDTO
- + delete(TolIDTO): void
- + findAll(void): ResponseEntity<List<TollDTO>>
- + findByld(String): ResponseEntity<TollDTO>

TollWebService

- tollService: ITollService

<ISubRunWebService

- + add(SubRunDTO): SubRunDTO
- + update(SubRunDTO): SubRunDTO
- + delete(SubRunDTO): void
- + findAll(void): ResponseEntity<List<SubRunDTO>>>
- + findById(String): ResponseEntity<SubRunDTO>



SubRunService

- subRunService: ISubRunService

<<Interface>>

IPassengerWebService

- + add(PassengerDTO): PassengerDTO
- + update(PassengerDTO): PassengerDTO
- + delete(PassengerDTO): void
- + findAll(void): ResponseEntity<List<PassengerDTO>>
- + findByld(String): ResponseEntity<PassengerDTO>
- + findByUserId(String): ResponseEntity<PassengerDTO>

PassengerWebService

- PassengerService: IPassengerService

<<Interface>>

IRunPriceWebService

- + add(RunPriceDTO): RunPriceDTO
- + update(RunPriceDTO): RunPriceDTO
- + delete(RunPriceDTO): void
- + findAll(void): ResponseEntity<List<RunPriceDTO>>>
- + findByld(String): ResponseEntity<RunPriceDTO>
- + findByPower(int): ResponseEntity<RunPriceDTO>



RunPriceWebService

- runPriceService: IRunPriceService

<<Interface>>

ITransactionWebService

- + add(TransactionDTO): TransactionDTO
- + update(TransactionDTO): TransactionDTO
- + delete(TransactionDTO): void
- + findAll(void): ResponseEntity<List<TransactionDTO>>
- + findByld(int): ResponseEntity<TransactionDTO>
- + findByTransactionNumber(String): ResponseEntity<TransactionDTO>
- + findByTransactionDate(DateTime): ResponseEntity<List<TransactionDTO>>>
- + findBySenderId(int): ResponseEntity<List<TransactionDTO>>>
- + findByReceiver(int): ResponseEntity<List<TransactionDTO>>
- + findByState(TxSTate): ResponseEntity<List<TransactionDTO>>>
- + findByTransactionDateAndSenderId(DateTime, String): ResponseEntity<List<TransactionDTO>>>
- + findByTransactionDateAndReceiverId(DateTime, String): ResponseEntity<List<TransactionDTO>>>
- + findBySenderIdAndReceiverId(String, String): ResponseEntity<List<TransactionDTO>>
- + findByTransactionDateAndState(DateTime, TxState): ResponseEntity<List<TransactionDTO>>



TransactionWebService

- transactionService: ITransactionService