

**<<Interface>>
IImageStorageService**

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

ImageStorageService

- imageStorageRepository: IImageStorageRepository

**<<Interface>>
IAccountWebService**

+ add(AccountDTO): AccountDTO
+ update(AccountDTO): AccountDTO
+ delete(String): 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>>
IMessageWebService**

+ add(MessageDTO): MessageDTO
+ update(MessageDTO): MessageDTO
+ delete(String): void
+ findAll(void): ResponseEntity<List<MessageDTO>>
+ findById(String): ResponseEntity<MessageDTO>
+ findByHoroLessThanEqual(LocalDateTime): ResponseEntity<List<MessageDTO>>
+ findByHoroGreaterThanEqual(LocalDateTime): ResponseEntity<List<MessageDTO>>
+ findByHoroBetween(LocalDateTime, LocalDateTime): ResponseEntity<List<MessageDTO>>
+ findBySenderId(String): ResponseEntity<List<MessageDTO>>
+ findBySenderIdAndHoroLessThanEqual(String, LocalDateTime): ResponseEntity<List<MessageDTO>>
+ findBySenderIdAndHoroGreaterThanEqual(String, LocalDateTime): ResponseEntity<List<MessageDTO>>
+ findBySenderIdAndHoroBetween(String, LocalDateTime, LocalDateTime): ResponseEntity<List<MessageDTO>>
+ findByReceiverId(String): ResponseEntity<List<MessageDTO>>
+ findByReceiverIdAndHoroLessThanEqual(String, LocalDateTime): ResponseEntity<List<MessageDTO>>
+ findByReceiverIdAndHoroGreaterThanEqual(String, LocalDateTime): ResponseEntity<List<MessageDTO>>
+ findByReceiverIdAndHoroBetween(String, LocalDateTime, LocalDateTime): ResponseEntity<List<MessageDTO>>
+ findBySenderIdAndReceiverId(String, String): ResponseEntity<List<MessageDTO>>
+ findBySenderIdAndReceiverIdAndHoroLessThanEqual(String, String, LocalDateTime): ResponseEntity<List<MessageDTO>>
+ findBySenderIdAndReceiverIdAndHoroGreaterThanEqual(String, String, LocalDateTime): ResponseEntity<List<MessageDTO>>
+ findBySenderIdAndReceiverIdAndHoroBetween(String, String, LocalDateTime, LocalDateTime): ResponseEntity<List<MessageDTO>>

MessageWebService

- messageService: IMessageService

**<<Interface>>
IRunWebService**

+ add(RunDTO): RunDTO
+ update(RunDTO): RunDTO
+ delete(String): void
+ findAll(void): ResponseEntity<List<RunDTO>>
+ findById(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>>
+ findPassedByUserId(String): ResponseEntity<List<RunDTO>>
+ findPassedNotCancelledByUserId(String): ResponseEntity<List<RunDTO>>
+ find(String, String, Date, String, String): ResponseEntity<List<RunDTO>>

RunWebService

- runService: IRunService

**<<Interface>>
IRunPriceWebService**

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

RunPriceWebService

- runPriceService: IRunPriceService

**<<Interface>>
IAddressWebService**

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

AddressWebService

- addressService: IAddressService

**<<Interface>>
IUserWebService**

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

UserWebService

- userService: IUserService

**<<Interface>>
ITransactionWebService**

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