Dokumentacja innego projektu

Endpoint	Description	Request body

/auth	Used for creating account.
	
	Password should be 8 characters and contain at least: 1 uppercase letter, 1 special sign, 1 number.
	
	ℹ️ HTTP METHOD: [POST]
	
	❌ 400 - Bad request - user already exists or password is too weak.
	
	✅ 201 - Created
		Request body:
		
		``json
		{
		    "login": "string",
		    "password": "S@molocik12345"
		}
		``
		
		Response body:
		
		``json
		{
		    "login": "string",
		    "role": "string",
		    "token": "string"
		}
		``

/auth/login	Used for logging in to account.
	
	ℹ️ HTTP METHOD: [POST]
	
	❌ 400 - Bad request - login not found or validation error
	
	✅ 200 - Created
		Request body:
		
		``json
		{
		    "login": "string",
		    "password": "string",
		}
		``
		
		Response body:
		
		``json
		{
		    "login": "string",
		    "role": "string",
		    "token": "string"
		}
		``

/auth/device/login	Used for logging in with device.
	
	Sending login, password and name **creates new device.**
	
	Sending login, password and uuid **logs device to the system.**
	
	ℹ️ HTTP METHOD: [POST]
	
	❌ 400 - Bad request - username not found, validation error or not linked device
	
	✅ 200 - Ok
		Request body:
		
		``json
		{
		    "login": "string",
		    "password": "string",
		    "uuid": "string",
		    "name": "string"
		}
		``
		
		Response body:
		
		``json
		{
		    "token": "string",
		    "uuid": "string"
		}
		``

/client	** NOT FINISHED **	

/device	Returns list of devices fullfiling specified search criteria.
	
	You can search by:
	
	- uuid,
	- device type:
	  - "IOS", "LINUX", "ANDROID"
	  - you can search by more than one device type at once,
	- device status:
	  - "ACTIVE", "WORKING", "INACTIVE",
	  - you can search by more than one device status at once,
	- device name,
	
	or sort by:
	
	- uuid descending - "UUID_DESC",
	- uuid ascending - "UUID_ASC",
	- name descending - "NAME_DESC",
	- name ascending - "NAME_ASC"
	
	Example request:
	
	<code>/device?
	deviceTypes=I
	OS&deviceType
	s=LINUX&devic
	eStatuses=WOR
	KING</code>
	
	ℹ️ HTTP METHOD: [GET]
	
	✅ 200 - Ok
		Params:
		
		"uuid", "deviceTypes", "deviceStatuses", "name", "sortBy"
		
		Response body:
		
		<pre>[
		  {
		    "uuid": "string",
		    "deviceType": "string",
		    "deviceModel": "string",
		    "lastConnection": "string",
		    "status": "string"
		  },
		  ...
		]</pre>

/device/tasks	Returns list of tasks for device specified by uuid.	**Request header:**
| | Device uuid must be passed in request header. | ```
headers: {
    "Content-Type": "json
/application",
    "Authorization": token,
    "device-uuid": uuid,
},
``` |
| | ℹ️ HTTP METHOD: [GET] | |
| | ❌ 400 - Bad request - invalid device type | **Response body:** |
| | ✅ 200 - Ok | ```
[
    {
     "id": "long",
     "taskType": "string",
     "status": "string",
     "fileUrl": "string",
     "orderStart": "string",
     "orderEnd": "string",
     "device": {
      "uuid": "string",
      " deviceType": "string",
      "deviceModel": "string",
     }
    },
    ...
]
``` |

| **device/tasks/ {taskId}/upload** | Uploads file. Example request: <br><br> ```
/tasks?
deviceTypes=I
OS&deviceType
s=LINUX&taskS
tatuses=FINIS
HED
``` <br><br> ℹ️ HTTP METHOD: [POST] <br><br> ❌ 400 - Bad request <br><br> ✅ 200 - Ok | **Request header:** <br><br> ```
headers: {
    "Content-Type": "multipart
/form-data",
    "Authorization": token,
    "device-uuid": uuid,
},
``` <br><br> **Path variable:** <br> "taskId" must be passed as path variable. <br><br> **Request body:** <br><br> ```
{
   "file": file,
}
``` |
| **/tasks** | Used for creating tasks. <br><br> Task types: <br> • "STREAMING", <br> • "SFTP", <br> • "HTTP" <br><br> ℹ️ HTTP METHOD: [POST] <br><br> ❌ 400 - Bad request <br><br> ✅ 200 - Ok | **Request body:** <br><br> ```
{
   "uuid": "string",
   "type": "string",
}
``` <br><br> **Response body:** <br><br> ```
{
   "id": "long",
   "taskType": "string",
   "status": "string",
   "fileUrl": "string",
   "orderStart": "string",
   "orderEnd": "string",
   "deviceGetDto": {
     "uuid": "string",
     "deviceType": "string",
     "deviceModel": "string",
   }
}
``` |

/tasks	Returns list of devices fullfiling speified search criteria.
	
	You can search by:
	
	• Task types:
	    • "STREAMING", "SFTP", "HTTP",
	    • you can search by more than one task type at once,
	• device uuid,,
	• device type:
	    • "IOS", "LINUX", "ANDROID",
	    • you can search by more than one device status at once,
	• task status:
	    • "NEW", "IN_PROGRESS", "FINISHED",
	    • you can search by more than one task status at once,
	• device name,
	
	Example request:
	
	`/tasks? deviceTypes=I OS&deviceType s=LINUX&taskS tatuses=FINIS HED`
	
	ℹ️ HTTP METHOD: [GET]
	
	❌ 400 - Bad request
	
	✅ 200 - Ok
		Params:
		
		"taskTypes", "deviceUuid", "deviceName", "taskStatuses", "deviceTypes"
		
		Response body:
		
		``
		[
		  {
		    "id": "long",
		    "taskType": "string",
		    "status": "string",
		    "fileUrl": "string",
		    "orderStart": "string",
		    "orderEnd": "string",
		    "device": {
		      "uuid": "string",
		      "deviceType": "string",
		      "deviceModel": "string",
		    }
		  },
		  ...
		]
		``

/tasks/{taskId} /download	Downloads file. ℹ️ HTTP METHOD: [GET] ❌ 400 - Bad request ✅ 200 - Ok	**Path variable:** "taskId" must be passed as path variable. **Response body:** You receive file.