Development Guide
Push Demo

Version: V2

Version History

Date	Version	Description	Author	Approved By
2015-09-16	0.1	Created the initial draft.	Liao Zuzheng	
2015-10-31	0.2	Modified the class diagram in the "PushSDK Protocol Processing Classes" section and deleted the description of the classes that are not implemented. Added data communication diagrams regarding device command generation and obtaining to the "getrequest" section. Deleted the protocol description from the "Protocol Function Design" section. For protocol details, refer to protocol documents.	Liao Zuzheng	
2015-10-31	0.3	Added a disclaimer.	Liao Zuzheng	

Contents

1. Document Purpose

This document describes the architecture and design ideas for the Push Web Demo to help developers better understand the Demo code. It does not introduce the commercial use of the Demo.

2. Disclaimer

- Service Scope

  This software is a demo of the PushSDK protocol. The subsequent customer support and service cover only protocol data parsing, command delivery, and data integrity. The code is included in the com.zk.pushsdk package.

  Other software service-related supports are provided only within the support scope for the current functions of the Demo. For additional requirements, customization fees are charged according to the marketing department's evaluation result.

- Database design limits

  This Demo database is designed by following the device-based principle, which stipulates that, in the database, one record is created for one device even though the records of the devices are identical. The records include user information, fingerprint data template information, face data template information, and the like.

  A database designed based on this principle generates a large amount of redundant data during tests. During software development, customers must adjust the principle according to their service logic.

  Due to data redundancy, the Demo supports only five devices with full data (user, fingerprint, face, and attendance data).

- Device Models and Versions
  - Version of the PushSDK Protocol implemented by the Demo: 2.2.14.
  - PushSDK firmware version of associated devices: 2.0.14.20151021 or later.

- Function List

  The Demo supports all the functions specified in the protocol document. For details, see the protocol document.

3. Development Environment

3.1 Basic Environment

For installation details of **Tomcat**, **MySQL**, and **JDK**, see the *Push Demo Deployment Guide*.

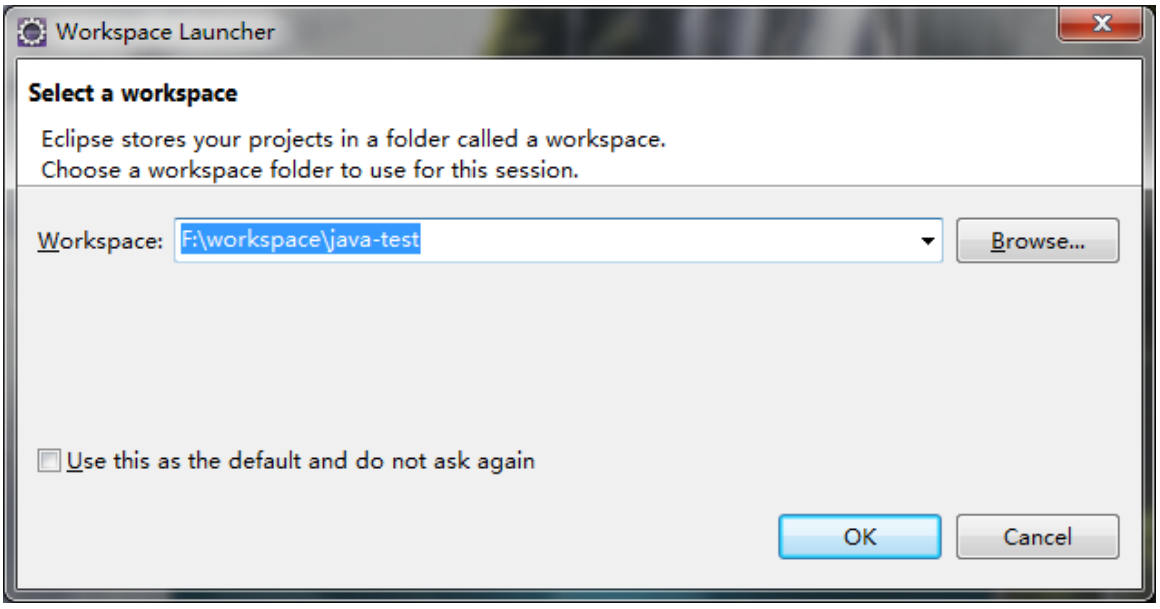
3.2 Configuring the Eclipse Environment

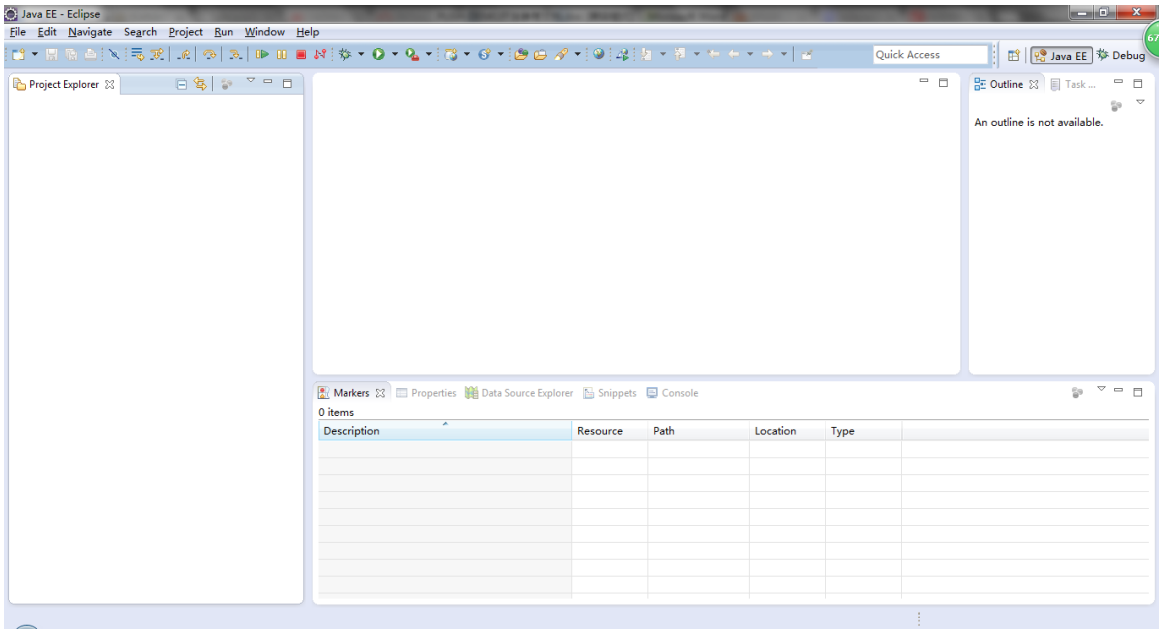
The Eclipse version used in this example is eclipse-jee-juno-SR1-win32. For details of how to configure the other versions, see the related official documents.

There are other configuration methods in addition to the one used in this example, which are not described I this document.

3.2.1 Starting Workspace

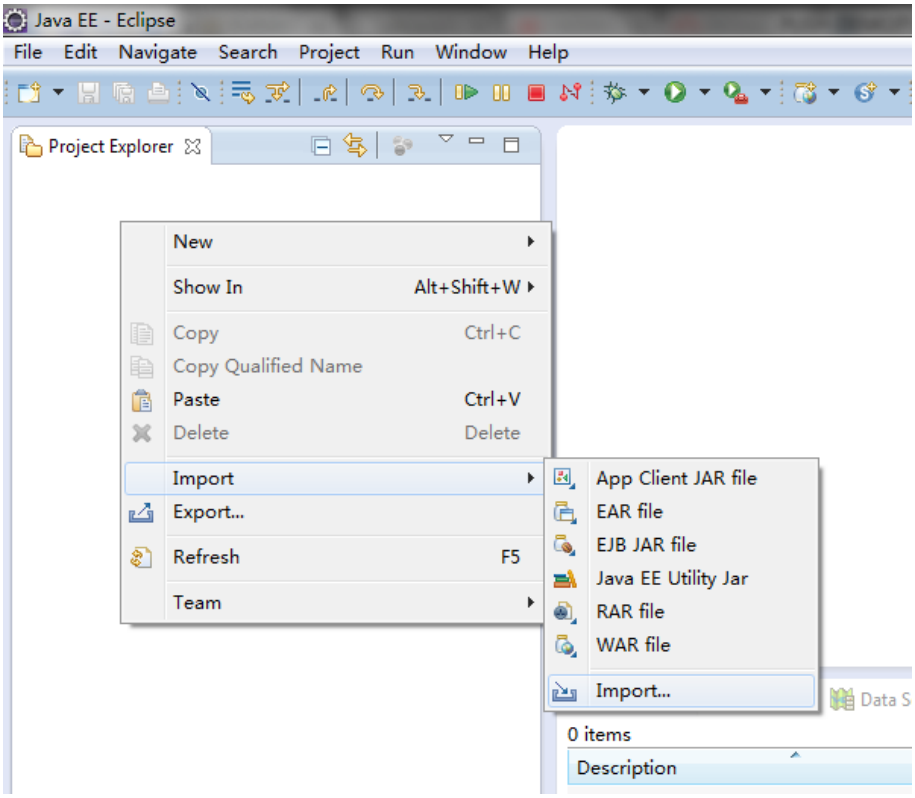
Start Eclipse, select Workspace, and click **OK**.



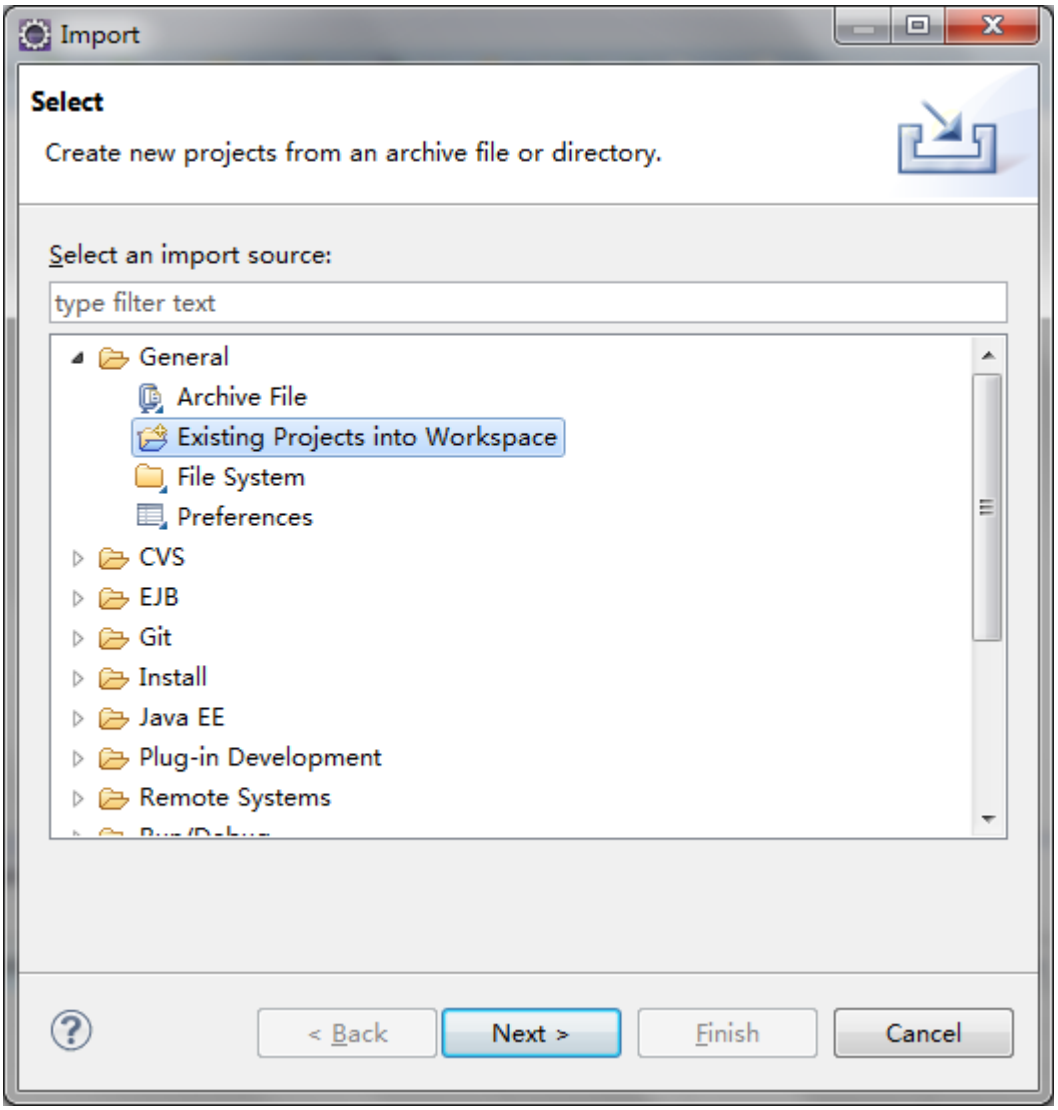


3.2.2 Imported a Project

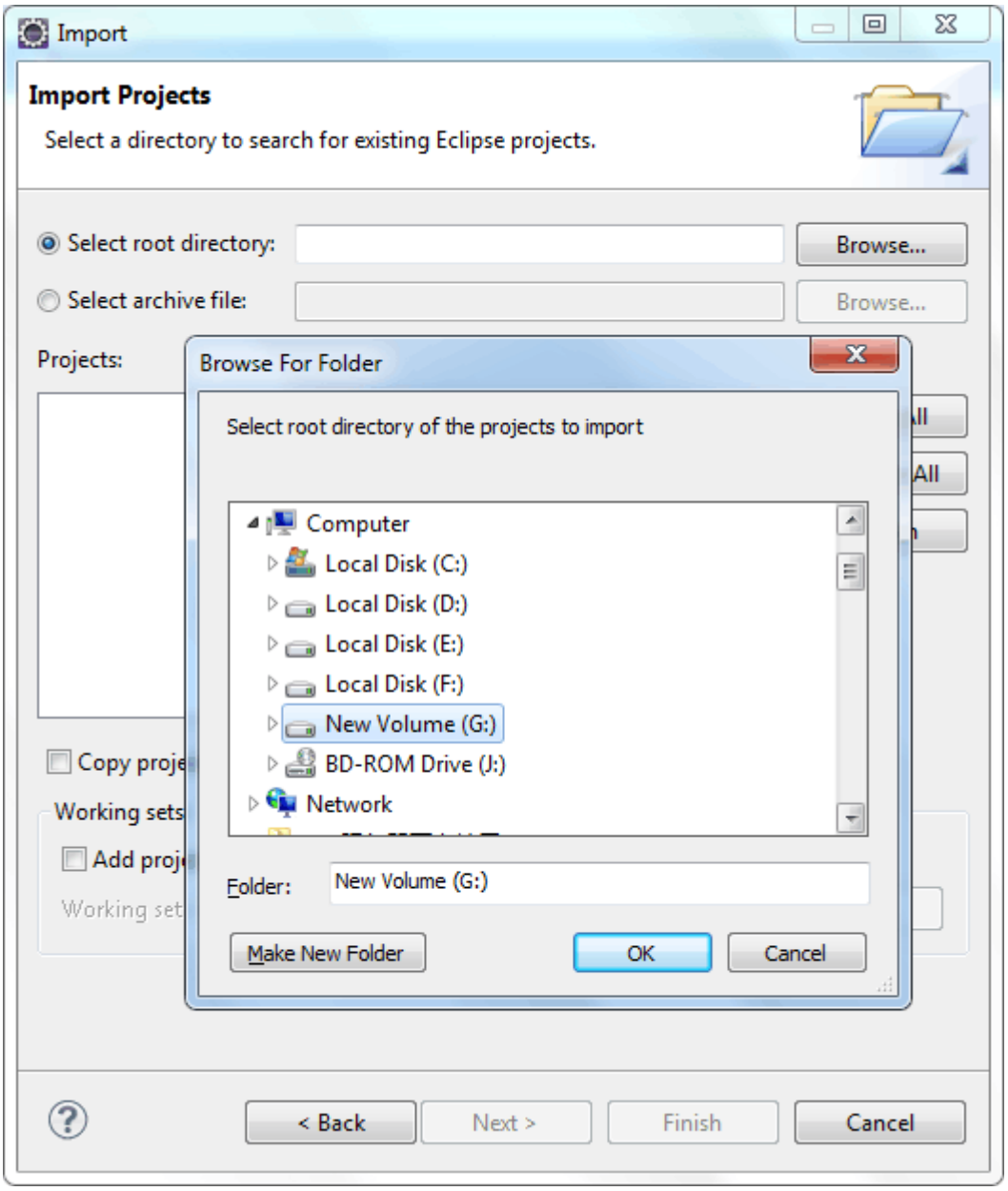
➢ Right-click in the Project Explorer pane and choose **Import** > **Import**.



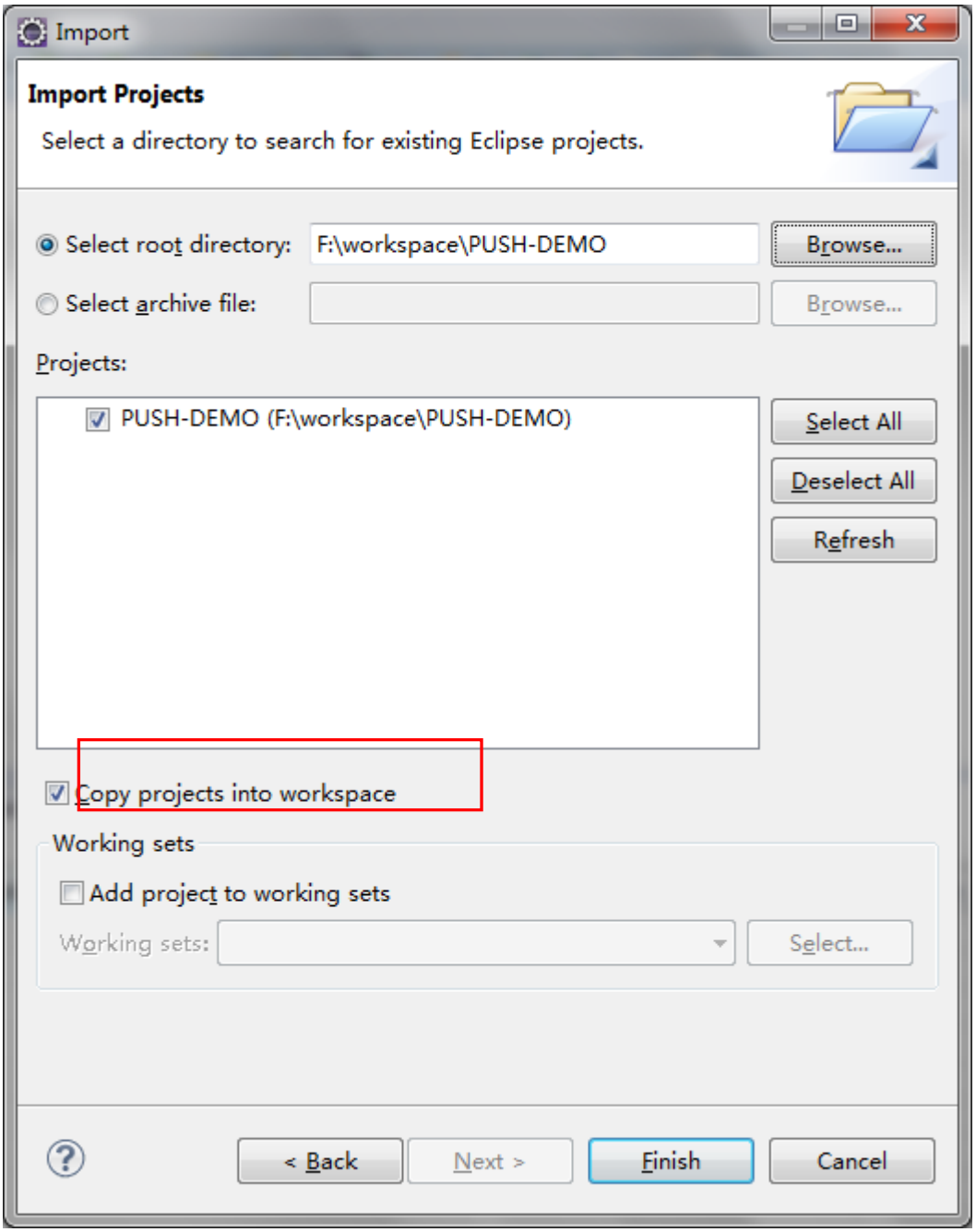
➢ Select Existing Projects into Workspace and click **Next**.

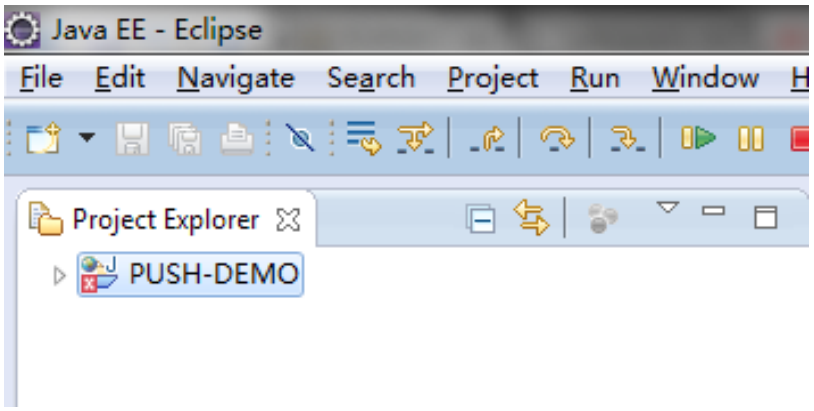


➢ Clock **Browse**, select the software project directory, and click **OK**.



➢ Click **Finish** to import the project. To copy the project to Workspace, select Copy projects into workspace highlighted in the red rectangle in the following below figure.

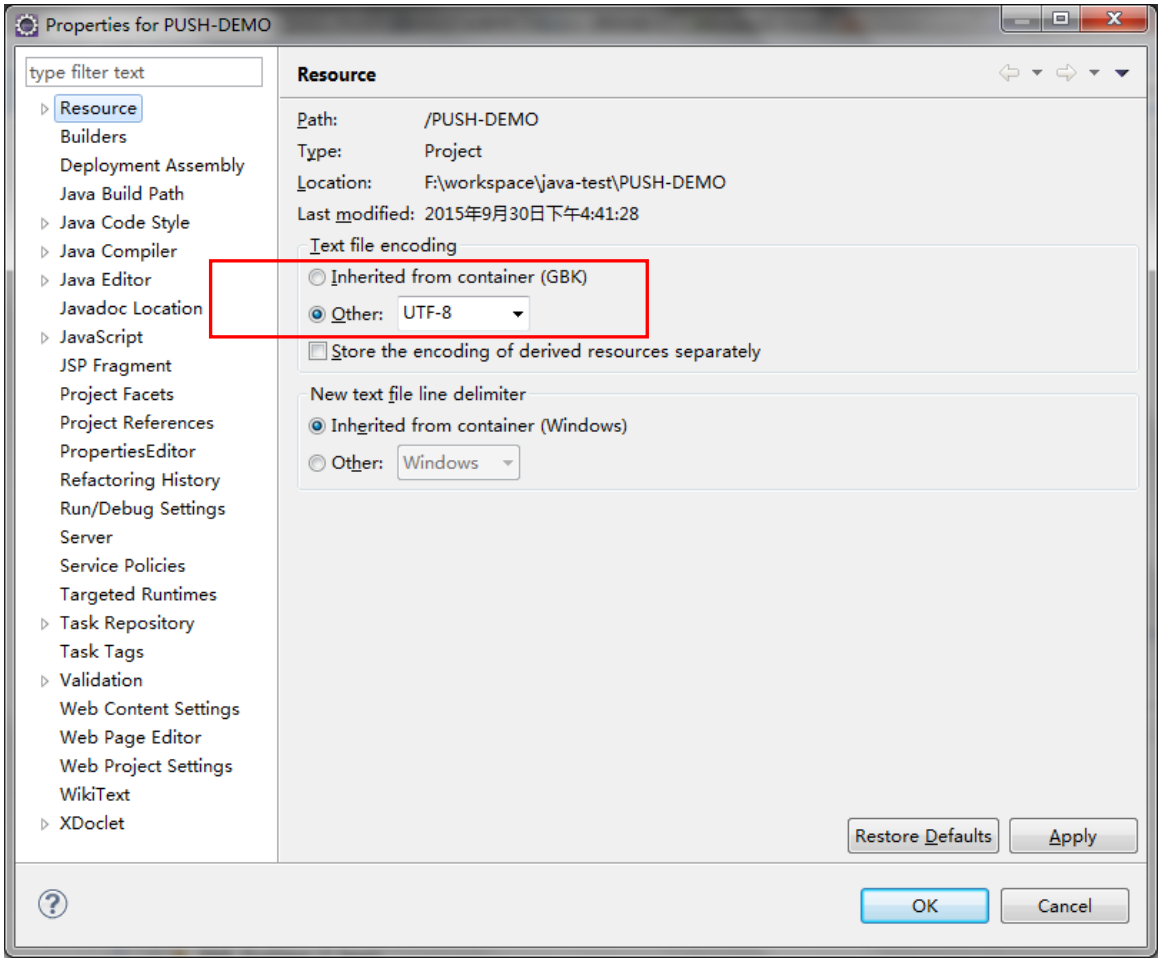

Import
Import Projects
Select a directory to search for existing Eclipse projects.
Select root directory:
F:\workspace\PUSH-DEMO
Browse...
Select archive file:
Browse...
Projects:
PUSH-DEMO (F:\workspace\PUSH-DEMO)
Select All
Deselect All
Refresh
Copy projects into workspace
Working sets
Add project to working sets
Working sets:
Select...
< Back
Next >
Finish
Cancel


Java EE - Eclipse
File Edit Navigate Search Project Run Window H
Project Explorer
PUSH-DEMO

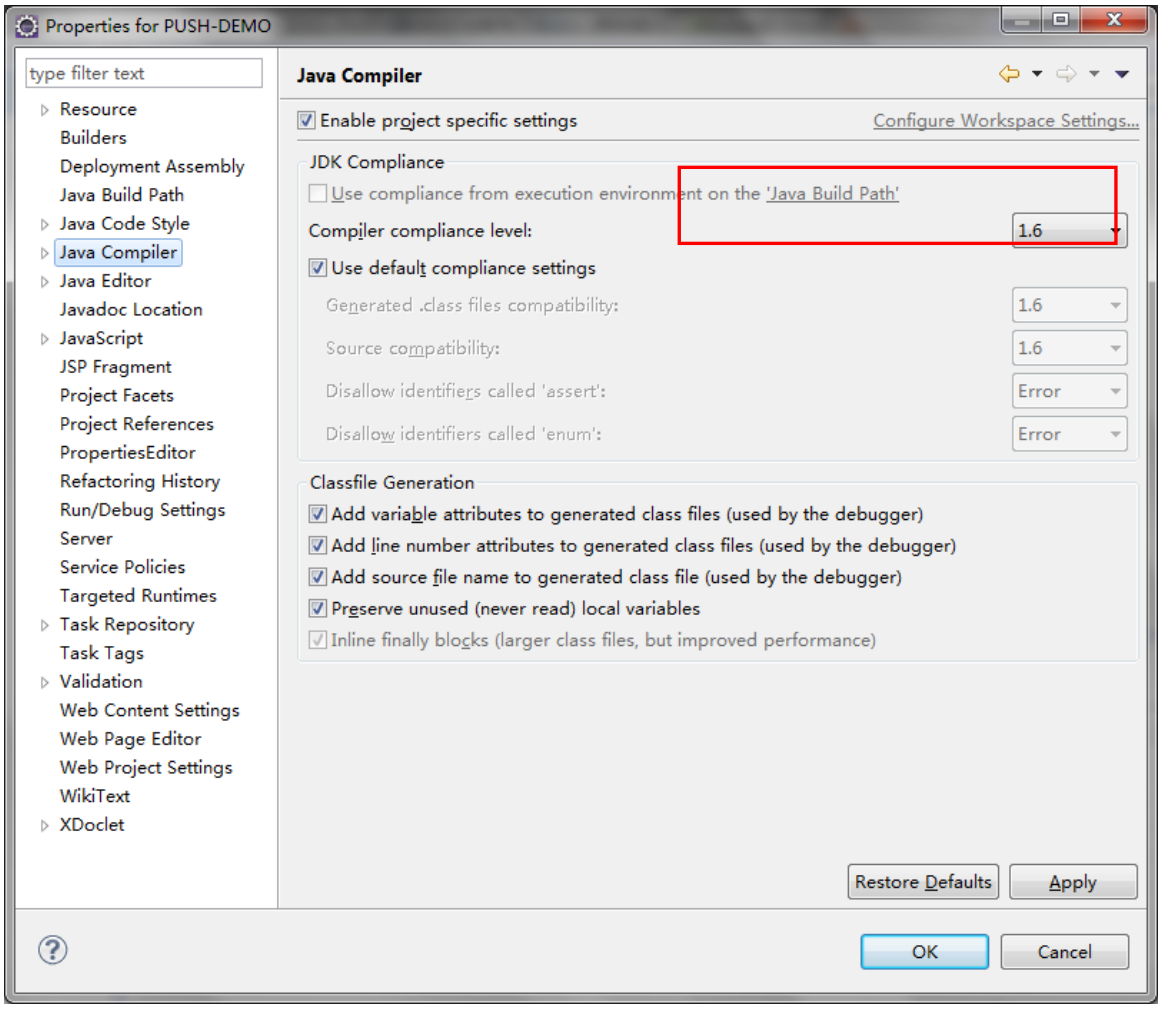
3.2.3 Setting Project Attributes

Right-click the project name in the Project Explorer pane and choose **Properties**.

- Project character encoding: **UTF-8**

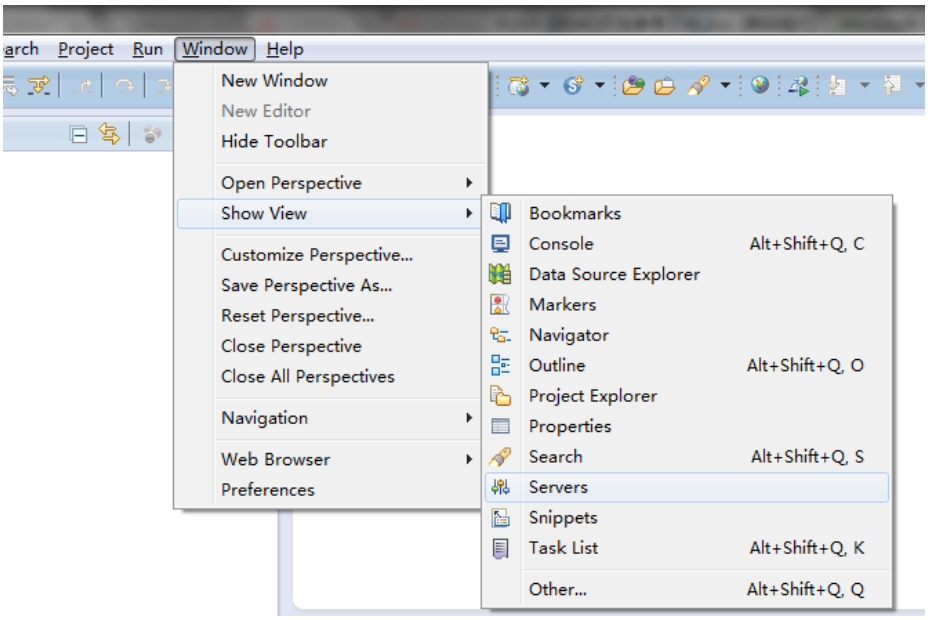


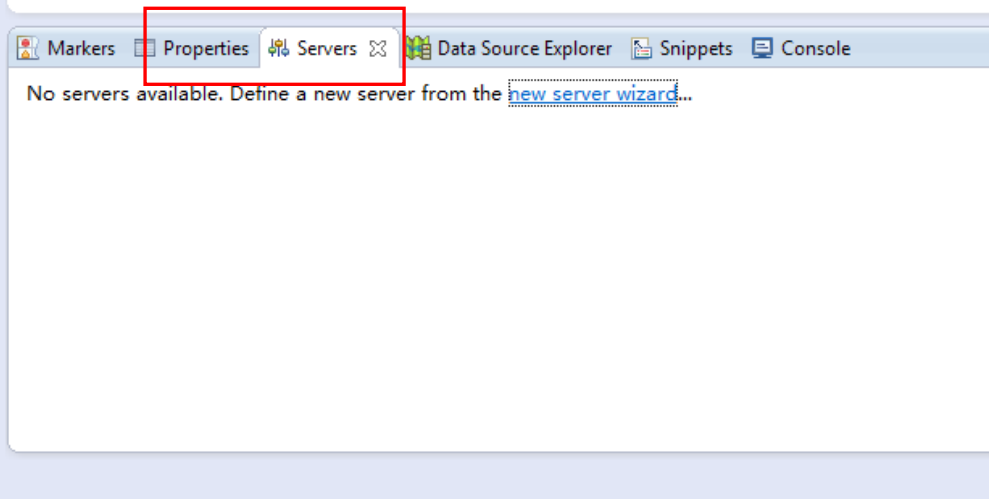
- Compiler version: 1.6



3.2.4 Configuring the Online Debugging Environment

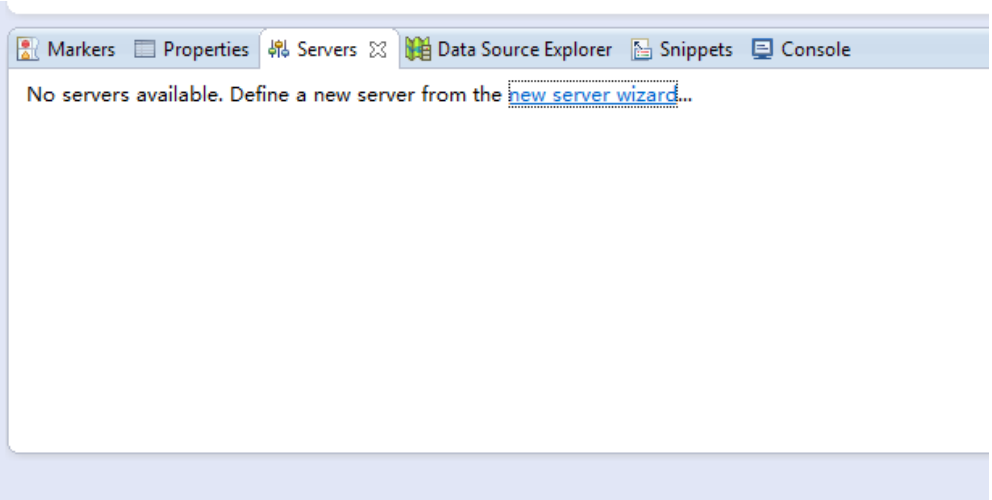
- Choose Window > Show View > Servers.



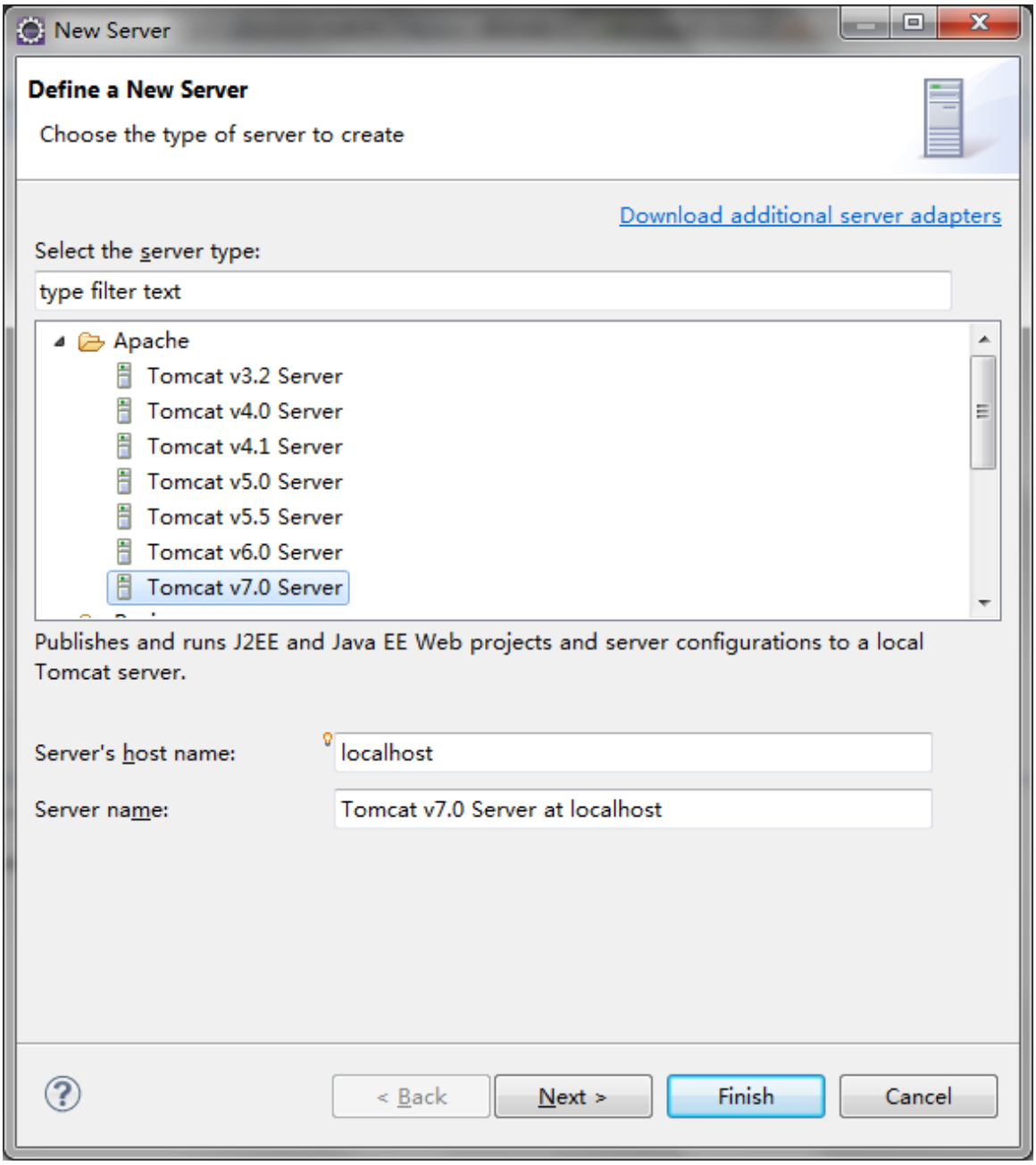


➢ Create a server instance, as follows:

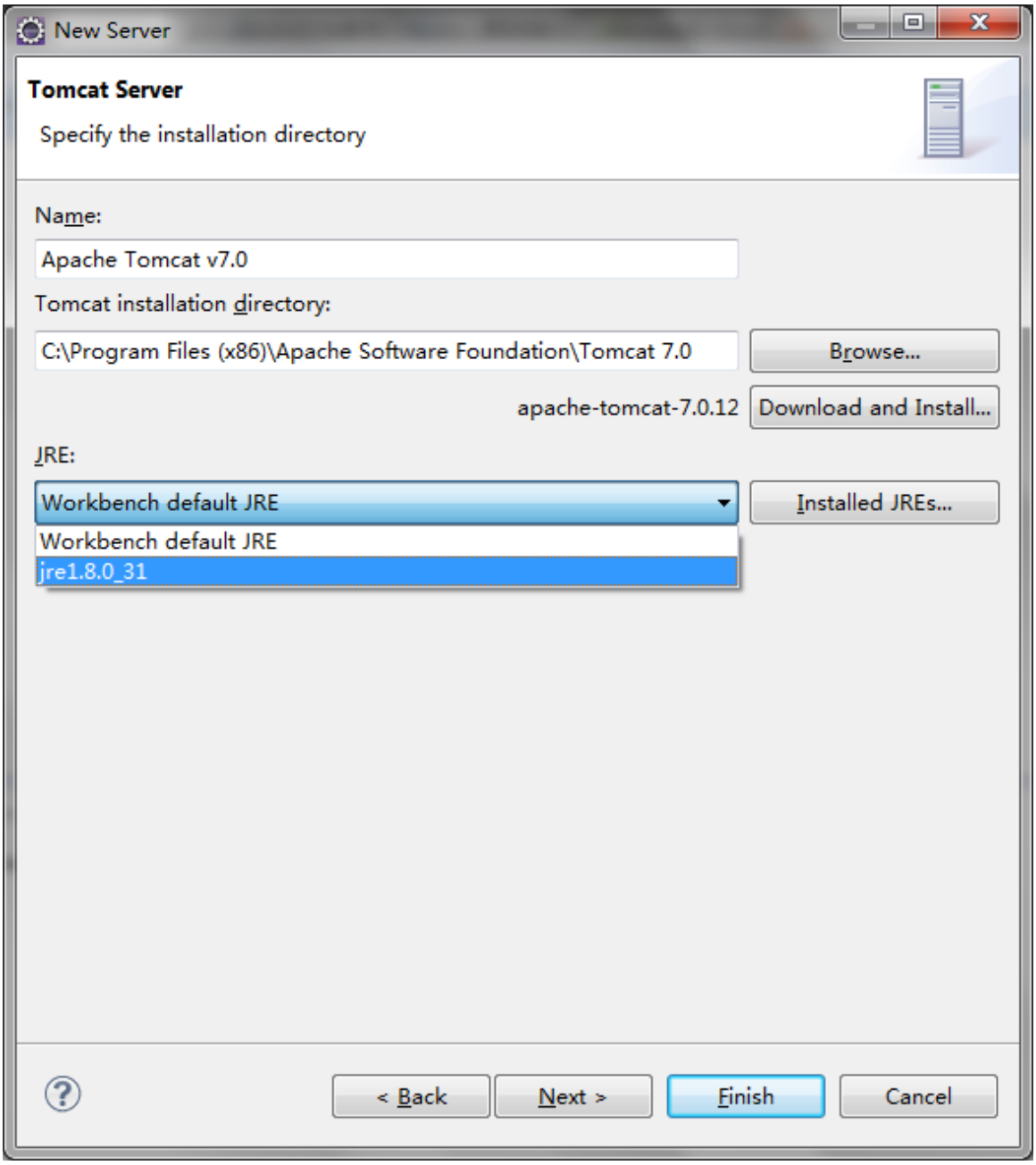
- On the Servers tab page, click **new server wizard**.



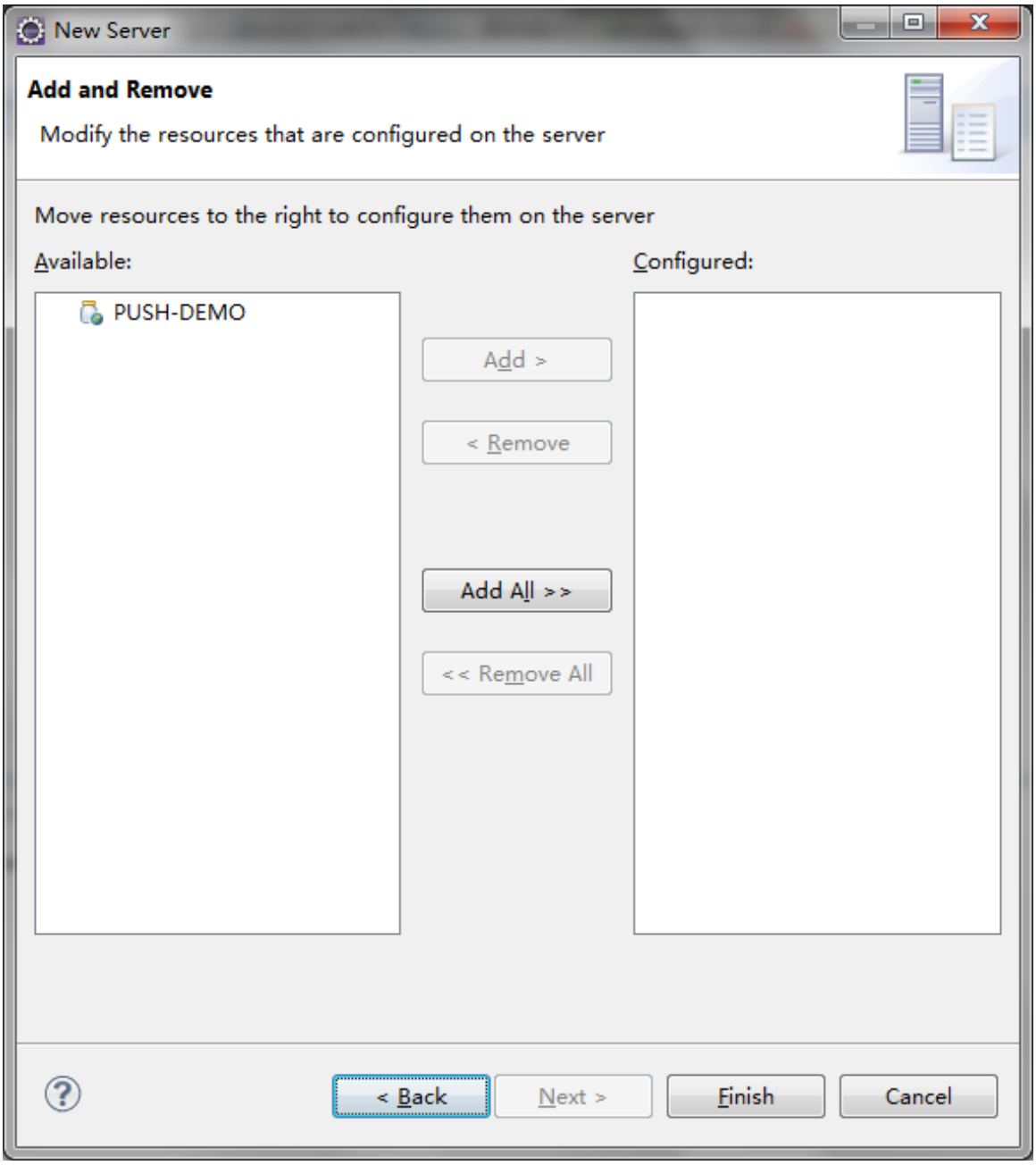
- Choose **Apache** > **Tomcat v7.0** Server and click **Next**.

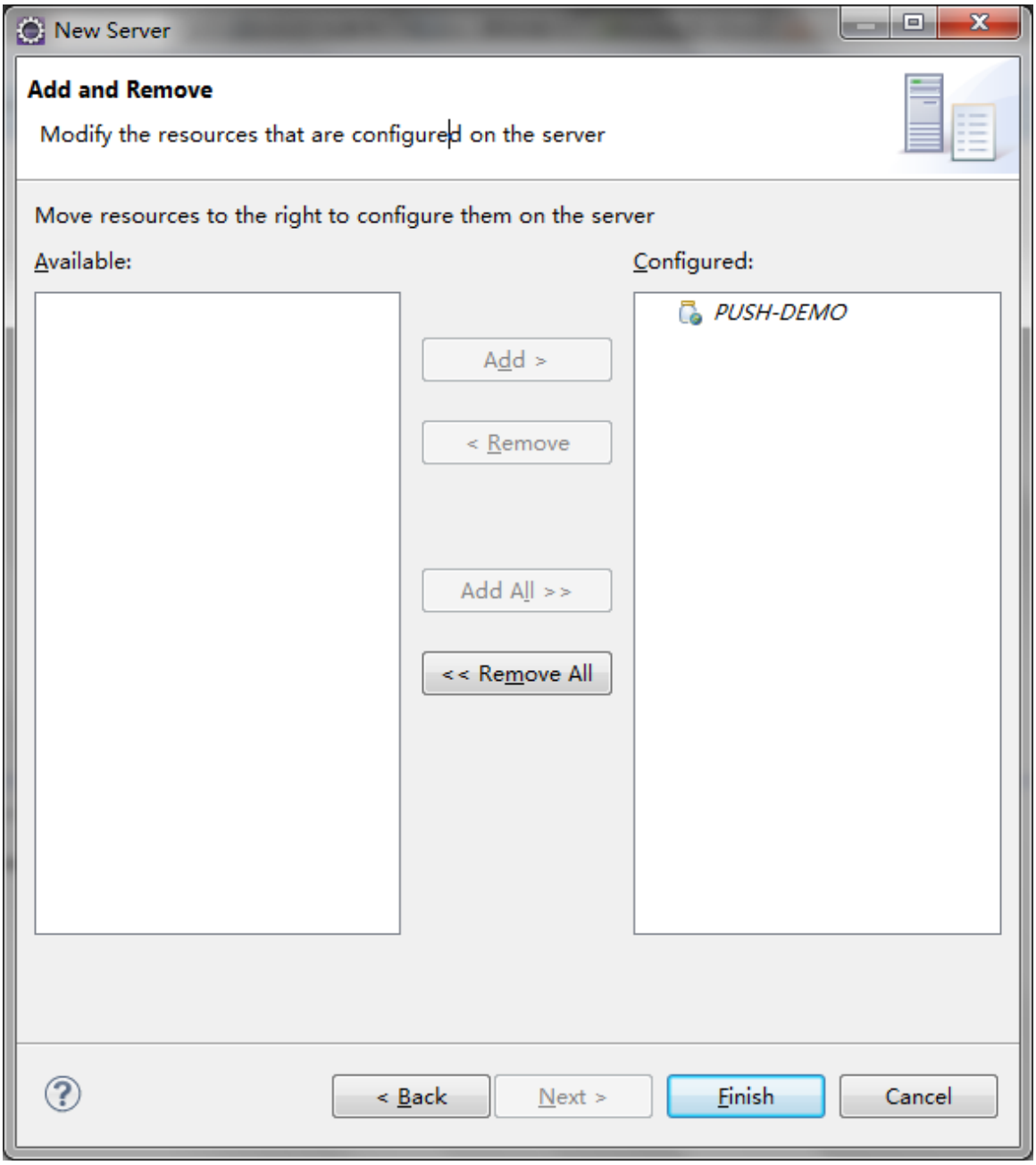


- Set JRE and click **Next**.



- Select the project on the left list, click **Add**> to add it to the right list, and click **Finish**.

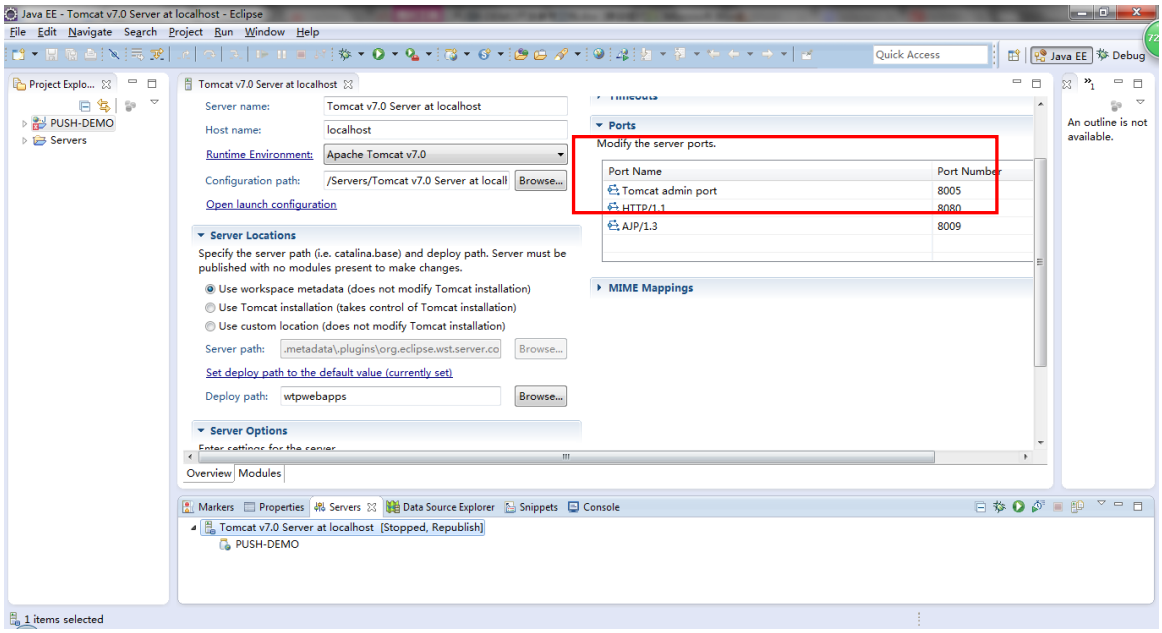

New Server
Add and Remove
Modify the resources that are configured on the server
Move resources to the right to configure them on the server
Available:
Configured:
PUSH-DEMO
Add >
< Remove
Add All >>
<< Remove All
< Back
Next >
Finish
Cancel



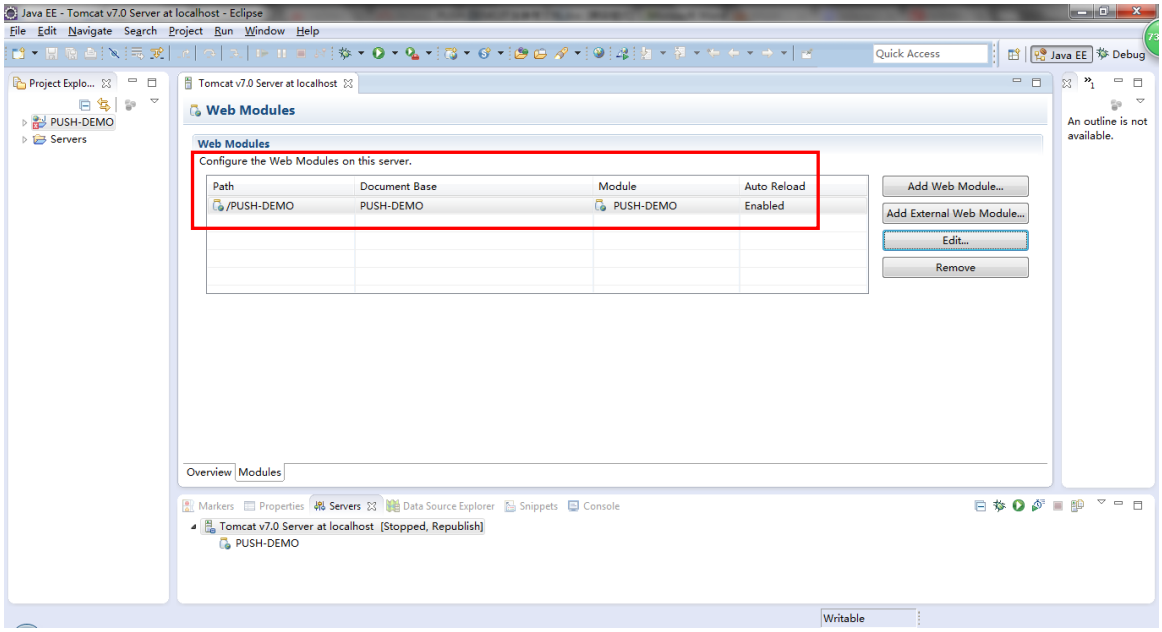
3.2.5 Configuring Project Deployment Information

➢ **Double-click** the online debugging server instance in the Servers window. The editing window appears.

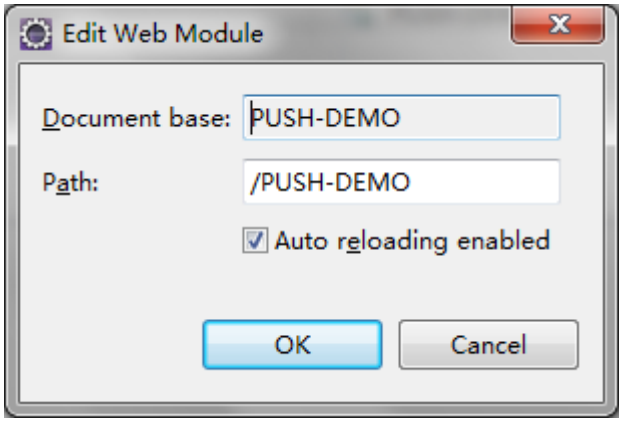
On the Overview tab page, set **web ports** as required.

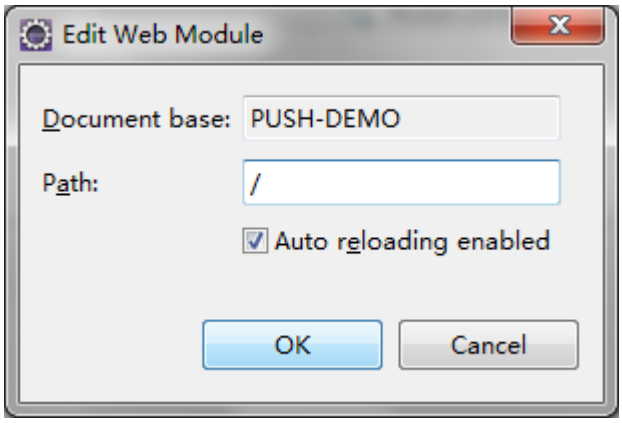


On the Modules tab page, select the **project**, and click **Edit**.



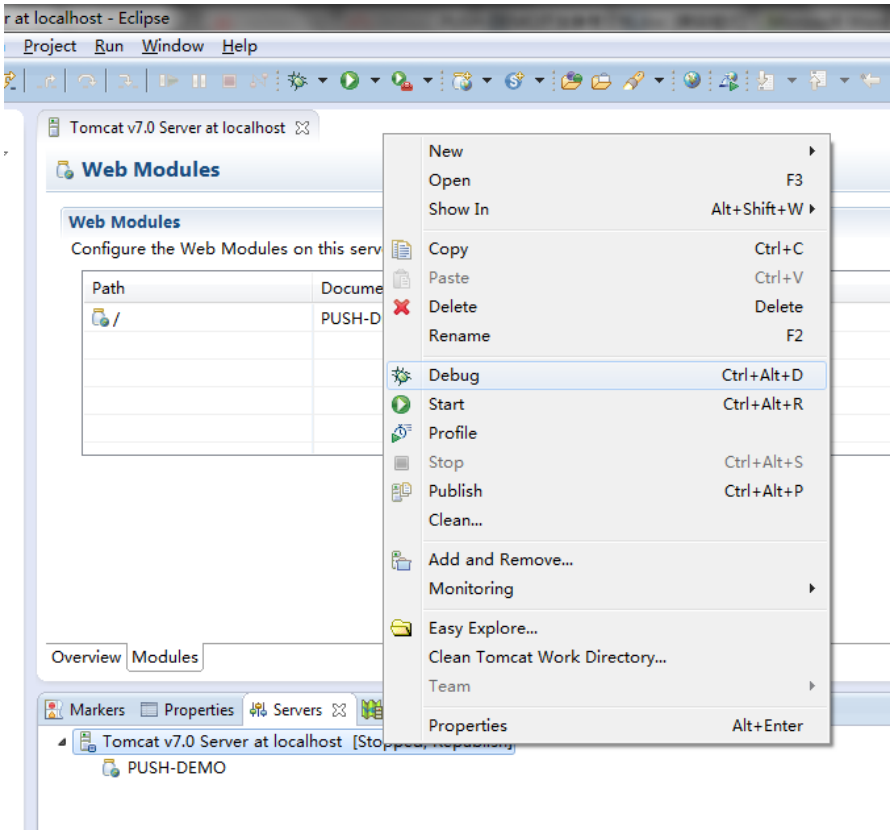
Set **Path** to / and click **OK**. See the following figure.



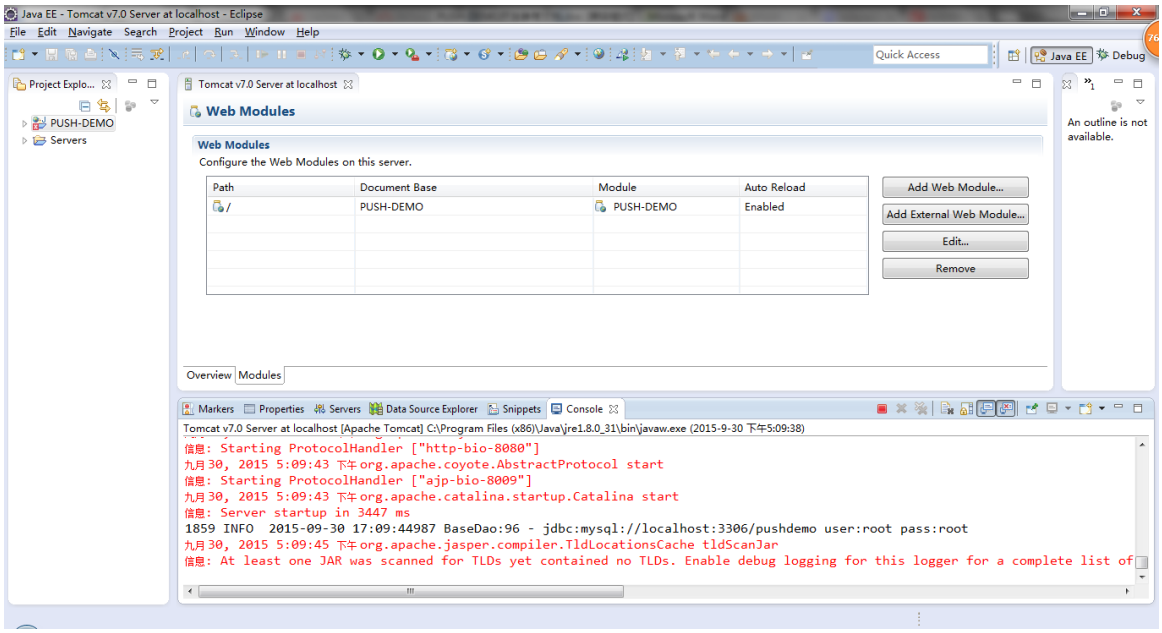


3.2.6 Starting the Server Instance

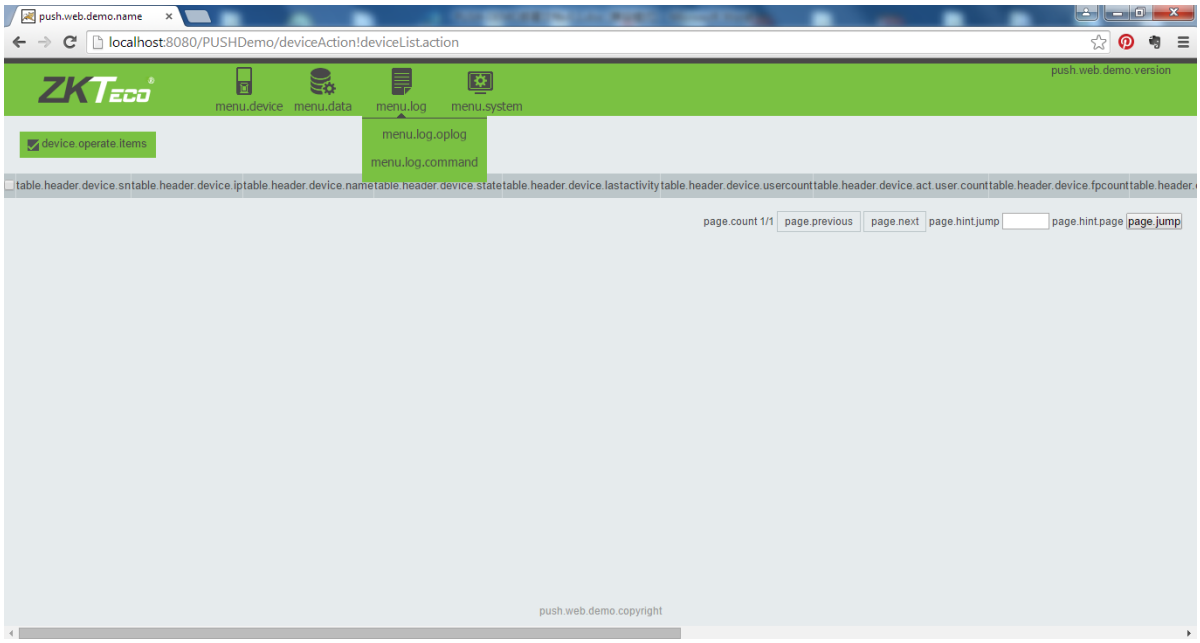
- In the Servers window, right-click the server instance and choose **Debug** to enter the Debug mode.



- You can view some output information on the Console tab page.



➢ Enter the server address on a web browser to check the server operating status.



4. Database Design

4.1 Design Ideas

According to the features of the Demo, the database tables adopt a device-based design, which means that, if a user work ID on a device is the same as a user work ID on another device, the two users are distinguished on the server.

Advantage: This facilitates data integrity test for different devices.

Disadvantage: This causes high data redundancy.

When developing a service system, customers can modify the database unique constraint and the **DAO** methods related to unique operation.

4.2 Database Table Structure

Database tables are stored using UTF-8.

4.2.1 Basic Device Information

Table	DEVICE_INFO		
Description	Basic device information		
Index	Type: unique; field: DEVICE_SN		
Field	Data Type [Length]	Constraint of Not Null	Description
DEVICE_ID	INT	Not null - primary key	Device ID, which increases automatically
DEVICE_SN	VARCHAR (24)	Not null - unique	Device SN
DEVICE_NAME	VARCHAR (30)	Not null	Device name
ALIAS_NAME	VARCHAR (24)		Device alias
DEPT_ID	INT		Department ID, which is not used currently
STATE	VARCHAR (30)	Not null	Device state, such as: Online Offline Communicating
LAST_ACTIVITY	VARCHAR (30)		Time when a device communicates with the server
TRANS_TIMES	VARCHAR (50)		Time when a client checks for and transfers new data
TRANS_INTERVAL	INT	Not null	The interval at which a client checks for and transfers new data
LOG_STAMP	VARCHAR (30)		Attendance log time stamp
OP_LOG_STAMP	VARCHAR (30)		Operation log time stamp
PHOTO_STAMP	VARCHAR (30)		Attendance photo time stamp
FW_VERSION	VARCHAR (40)		Push process version number of a device

USER_COUNT	INT		Number of device users
FP_COUNT	INT		Number of fingerprints recorded by a device
TRANS_COUNT	INT		Number of attendance logs on a device
FP_ALG_VER	VARCHAR (20)		Fingerprint identification algorithm version number of a device
PUSH_VERSION	VARCHAR (20)		Push protocol version number of a device
IPADDRESS	VARCHAR (30)	Not null	Device IP address
DEV_LANGUAGE	VARCHAR (20)		Device language
PUSH_COMM_KEY	VARCHAR (30)		Push communication password
FACE_COUNT	INT		Number of face records on a device
FACE_ALG_VER	VARCHAR (30)		Face identification algorithm of a device
REG_FACE_COUNT	INT		Number of templates that a device uses to create a piece of face data
DEV_FUNS	VARCHAR (30)		Functions supported by a device
TRANS_FLAG	VARCHAR (200)		Flag transferred from the server to a device to notify the device of the data that must be uploaded to the server
ERROR_DELAY	INT		Device timeout interval
DELAY	INT		Heartbeat interval for a device to connect to the server
REAL_TIME	INT		Whether data is uploaded from a device in real time 1: yes 0: no
ENCRYPT	INT		Whether the data uploaded from a device is encrypted

4.2.2 Basic User Information

Table	USER_INFO		
Description	Basic user information		
Index	Type: unique; field: USER_PIN and DEVICE_SN		
Field	Data Type [Length]	Constraint of Not Null	Description
USER_ID	INT	Not null - primary key	User ID, which increases automatically
USER_PIN	VARCHAR(24)	Not null	User work ID
DEVICE_SN	VARCHAR (30)	Not null	Device SN

PRIVILEGE	INT		User privilege code
NAME	VARCHAR (24)		User name
PASSWORD	VARCHAR (16)		Password
FACE_GROUP_ID	INT		Face group ID, which is not used currently
ACC_GROUP_ID	INT		Access control group ID
DEPT_ID	INT		Department ID, which is not used currently
IS_GROUP_TZ	INT		Whether to use group time segments
VERIFY_TYPE	INT		Verification type code
MAIN_CARD	VARCHAR (30)		User's primary card ID
VICE_CARD	VARCHAR (30)		User's secondary card ID, which is not used currently
EXPIRES	INT		User account expiration date, which is not used currently
TZ	VARCHAR (30)		ID of the used time segment of a user
PHOTO_ID_NAME	VARCHAR (50)		User photo name
PHOTO_ID_SIZE	INT		User photo size (Base64)
PHOTO_ID_CONTENT	LONGTEXT		User photo data (Base64)

4.2.3 User Biometric Identification Template Information

Table	PERS_BIO_TEMPLATE		
Description	User biometric identification template information		
Index	Type: unique; field: USER_ID, USER_PIN, TEMPLATE_NO, BIO_TYPE, and DEVICE_SN		
Field	Data Type [Length]	Constraint of Not Null	Description
ID	INT	Not null - primary key	Biometric identification template ID, which increases automatically
USER_ID	INT	Not null	User ID, referenced from the primary key of user basic information
USER_PIN	VARCHAR (30)	Not null	User work ID
DEVICE_SN	VARCHAR (30)	Not null	Device SN
VALID	INT	Not null	Validity flag 0: invalid 1: valid Default value: 1
IS_DURESS	INT	Not null	Duress flag 0: no 1: yes Default value: 0

BIO_TYPE	INT	Not null	Biometric identification type 0: common 1: fingerprint 2: face 3: voiceprint 4: iris 5: retina 6: palm print 7: finger vein
VERSION	VARCHAR (24)	Not null	Biometric identification algorithm version Fingerprint algorithm versions: 9.0 and 10.0 (10.0 is the mainstream version, while a few customers use 9.0.) Face identification algorithm versions: 5.0 and 7.0 (7.0 is the mainstream version.) Finger vein identification algorithm version: 3.0
DATA_FORMAT	INT	Not null	Template format 0: ZK (default) 1: ISO 2: ANSI
TEMPLATE_NO	INT	Not null	Biometric type ID [Fingerprint] IDs: 0 - 9 0: little finger (left hand) 1: ring finger (left hand) 2: middle finger (left hand) 3: index finger (left hand) 4: thumb (left hand) 5: thumb (right hand) 6: index finger (right hand) 7: middle finger (right hand) 8: ring finger (right hand) 9: little finger (right hand) [finger vein] Same as those for fingerprint [Face] 0 [Iris] 0: left eye 1: right eye

			[Palm vein] 0: left hand 1: right hand The default value is 0 unless otherwise stipulated.
TEMPLATE_NO_INDEX	INT	Not null	Biometric type template ID If one fingerprint is recorded in multiple templates, the IDs begins from 0.
SIZE	INT	Not null	Data size of a biometric identification template
TEMPLATE_DATA	LONGTEXT	Not null	Data of a biometric identification template

4.2.4 Device Commands

Table	DEVICE_COMMAND		
Description	Device commands		
Index	Type: unique; field: DEVICE_SN and CMD_RETURN		
Field	Data Type [Length]	Constraint of Not Null	Description
DEV_CMD_ID	INT	Not null - primary key	Device command ID, which increases automatically It is also sent to devices.
DEVICE_SN	VARCHAR (24)	Not null	Device SN
CMD_CONTENT	LONGTEXT	Not null	Command content
CMD_COMMIT_TIMES	VARCHAR (30)		Command generation time
CMD_TRANS_TIMES	VARCHAR (30)		Command request time of a device
CMD_OVER_TIME	VARCHAR (30)		Time when a device notifies the server after it completes command execution
CMD_RETURN	VARCHAR (11)		Command return value
LONGTEXT	LONGTEXT		Result data (text) returned by a command

4.2.5 Attendance Logs

Table	ATT_LOG		
Description	Attendance logs		
Index	Type: unique; field: USER_PIN, VERIFY_TIME, DEVICE_SN, and VERIFY_TYPE		
Field	Data Type [Length]	Constraint of Not Null	Description
ATT_LOG_ID	INT	Not null - primary key	Attendance log ID, which increases automatically
USER_PIN	VARCHAR (24)	Not null	User work ID

VERIFY_TYPE	INT	Not null	Verification type, such as fingerprint, face, and card
VERIFY_TIME	VARCHAR (30)	Not null	Verification time
STATUS	INT	Not null	Attendance status, such as check-in and check-out
WORK_CODE	INT		Work code
SENSOR_NO	INT		Sensor ID, which indicates a local sensor or an external fingerprint scanner. It is not used currently.
ATT_FLAG	INT		Attendance flag, which indicates normal attendance, attendance out of specified time segments, or anti-passback. It is not used currently.
DEVICE_SN	VARCHAR (24)	Not null	Device SN
RESERVED1	INT		Reserved field 1
RESERVED2	INT		Reserved field 2

4.2.6 Attendance Photos

Table	ATT_PHOTO		
Description	Attendance photos		
Index			
Field	Data Type [Length]	Constraint of Not Null	Description
ID	INT	Not null - primary key	Attendance photo ID, which increases automatically
DEVICE_SN	VARCHAR (30)		Device SN
FILE_NAME	VARCHAR (50)		Attendance photo file name
SIZE	INT		File size
CMD	VARCHAR (30)		File uploading mode
FILE_PATH	VARCHAR (1024)		Absolute path to attendance photos stored on the server

4.2.7 Device Operation Logs

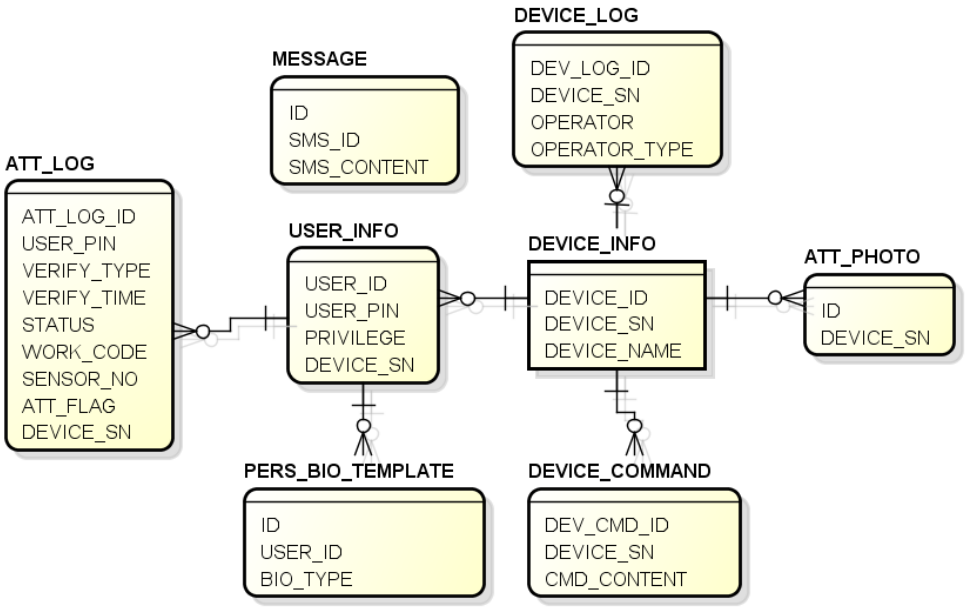
Table	DEVICE_LOG		
Description	Device operation logs		
Index			
Field	Data Type [Length]	Constraint of Not Null	Description
DEV_LOG_ID	INT	Not null -	Device operation log ID, which increases

		primary key	automatically
DEVICE_SN	VARCHAR (24)		Device SN
OPERATOR	VARCHAR (24)		Operator
OPERATOR_TYPE	INT		Operator type
OP_TYPE	INT		Operation type
OP_TIME	VARCHAR (30)		Operation time
VALUE1	VARCHAR (30)		Value 1
VALUE2	VARCHAR (30)		Value 2
VALUE3	VARCHAR (30)		Value 3
RESERVED	VARCHAR (30)		

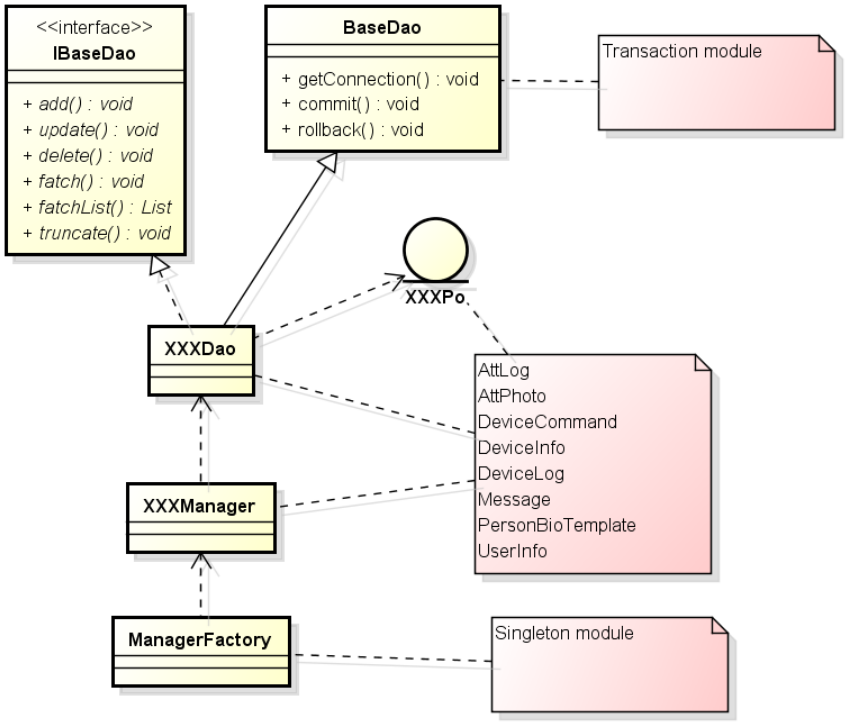
4.2.8 SMS Messages

Table	MESSAGE		
Description	SMS messages		
Index			
Field	Data Type [Length]	Constraint of Not Null	Description
ID	INT	Not null - primary key	SMS message ID, which increases automatically It is also sent to devices.
DEVICE_SN	VARCHAR (24)		Device SN, which is not used currently
SMS_TYPE	INT	Not null	SMS message type ➢ Public SMS message ➢ Private SMS message
START_TIME	VARCHAR (30)	Not null	SMS message effective time
VALID_MINUTES	INT	Not null	SMS message validity period (unit: minute)
SMS_CONTENT	VARCHAR (640)	Not null	SMS message content Single-byte character set: 320. Unicode character set: 160

4.3 ER Diagram



4.4 Database Module Class Diagram

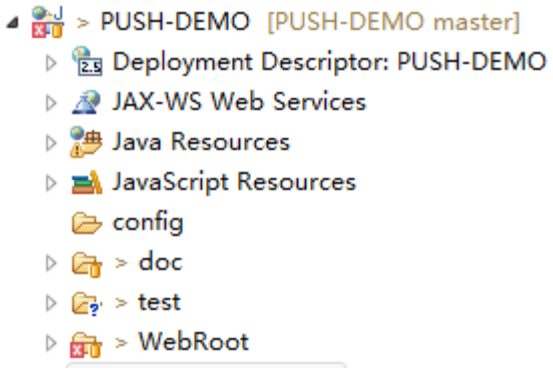


- IbaseDao: Provides basic method definition for the database table data operation.
- BaseDao: Provides methods related to database connection and provides database connections in transaction mode.
- XXXDao: Provides data table DAO implementation classes, which implement the methods of IbaseDao and special operation methods of specific tables.

- **XXXManager**: Provides data table management classes, which manage and combine database operation methods based on service requirements to obtain service data.
- **ManagerFactory**: Provides management factory classes, which provide specific database management class instances for service modules in single-instance mode.

5. Code Framework Design

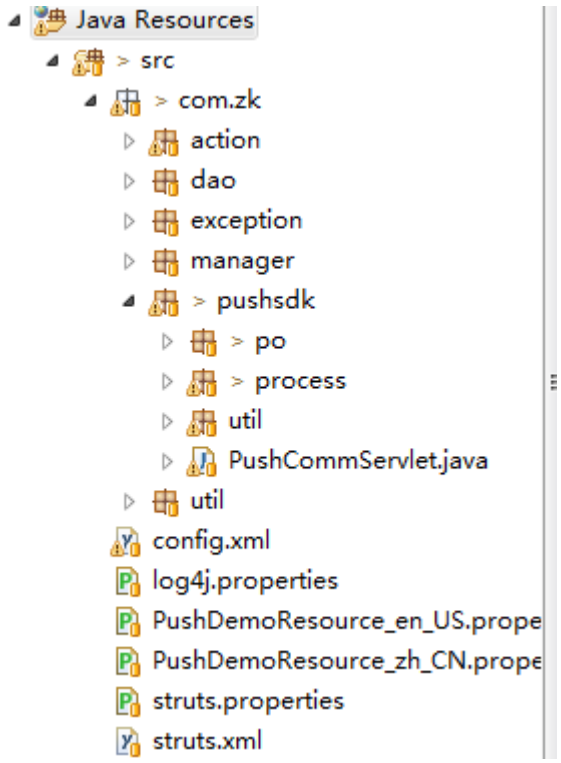
The following figure shows the code hierarchy.



- **Java source code**: Java Resources
- **Web code**: WebRoot

5.1 Java Source Code

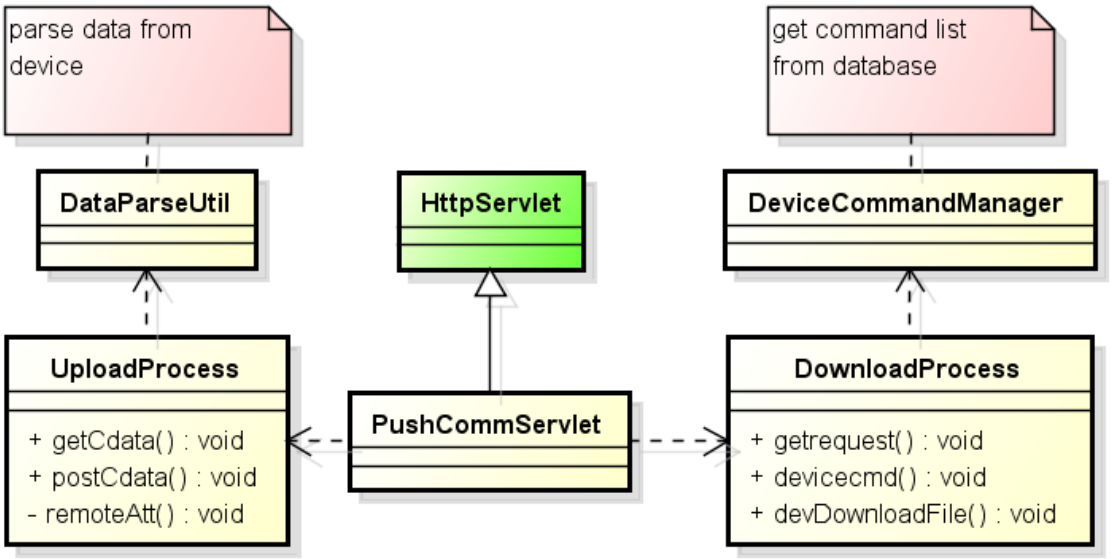
The following figure shows the Java source code package hierarchy.



- **action**: Classes that process page requests.
- **dao**: Classes that process database operations.
- **exception**: User-defined exception classes.
- **manager**: Service data management classes.
- **pushsdk**: Push protocol processing classes.
  - **po**: Data table instance objects.
  - **process**: Push protocol request processing classes.
  - **util**: Push protocol-related tool classes, such as data parsing and command assembling classes.
- **util**: Demo service tool classes, such as logging and paging classes.
- **config.xml**: Program parameter configuration file.
- **logj.properties**: Log configuration file.
- **PushDemoResource_xx_yy.properties**: Language file, where *xx* indicates the language code and *yy* indicates the country code.
  The combination of a language code and a country code indicates a language used in a country.
- **struts.properties**: Struts2 property configuration file for setting the multi-language properties.
- **struts.xml**: File for managing action mapping for the Demo.

5.1.1 PushSDK Protocol Processing Classes

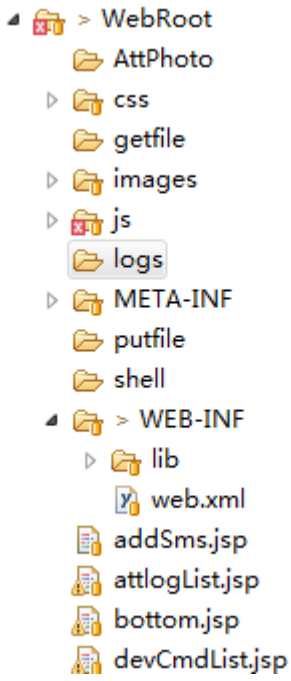
Protocol processing class diagram:



5.2 Web Code

5.2.1 WebRoot Directory

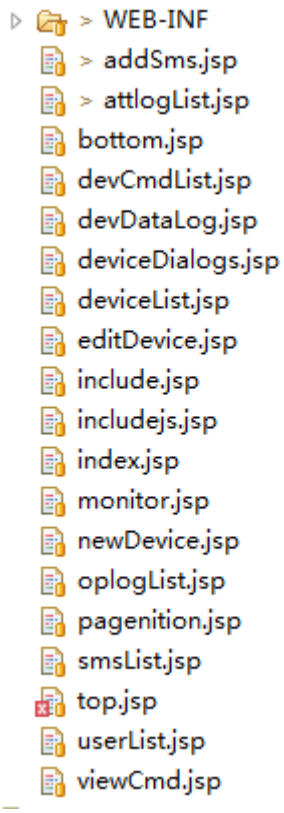
The following figure shows the page code hierarchy.



- **AttPhoto**: Stores attendance photos for the Demo.
- **css**: Store cascading style sheets for the Demo.
- **getfile**: Stores the files obtained from devices for the Demo.
- **images**: Stores the images used on the Demo UI.
- **js**: Stores the javascripts used on the Demo UI.
- **logs**: Stores Demo log information.
- **putfile**: Stores the files to be sent to devices for the Demo.
  Users can create other directories for this purpose.
- **shell**: Stores the results returned after shell-related commands (commands that generate text content) are executed by the Demo.
- **WEB-INF**: Web application security directory accessible only to the server.
  - lib: stores the jar packages used by the Demo (except JRE standard jar packages).
  - web.xml: program deployment description file.
  - *.jsp: page files

5.3 JSP Files

The Demo JSP files form the page presentation layer for logical data. See the following figure.



Public pages:

- **bottom.jsp**: Bottom page.
- **include.jsp**: Tab library page.
- **includes.jsp**: Javascript library page.
- **top.jsp**: Top page, which includes the main function menu for all pages.
- **deviceDialogs.jsp**: Pop-up javascript method page.

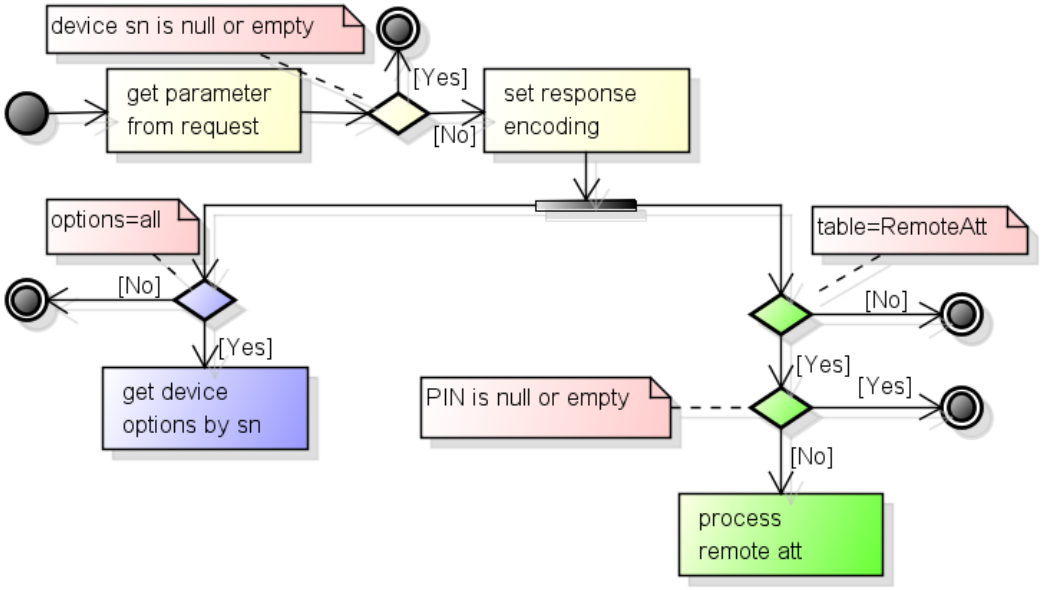
6. Protocol Function Design

6.1 Protocol Flowcharts

6.1.1 GET cdata

The operations that use the GET cdata method in server requests include:

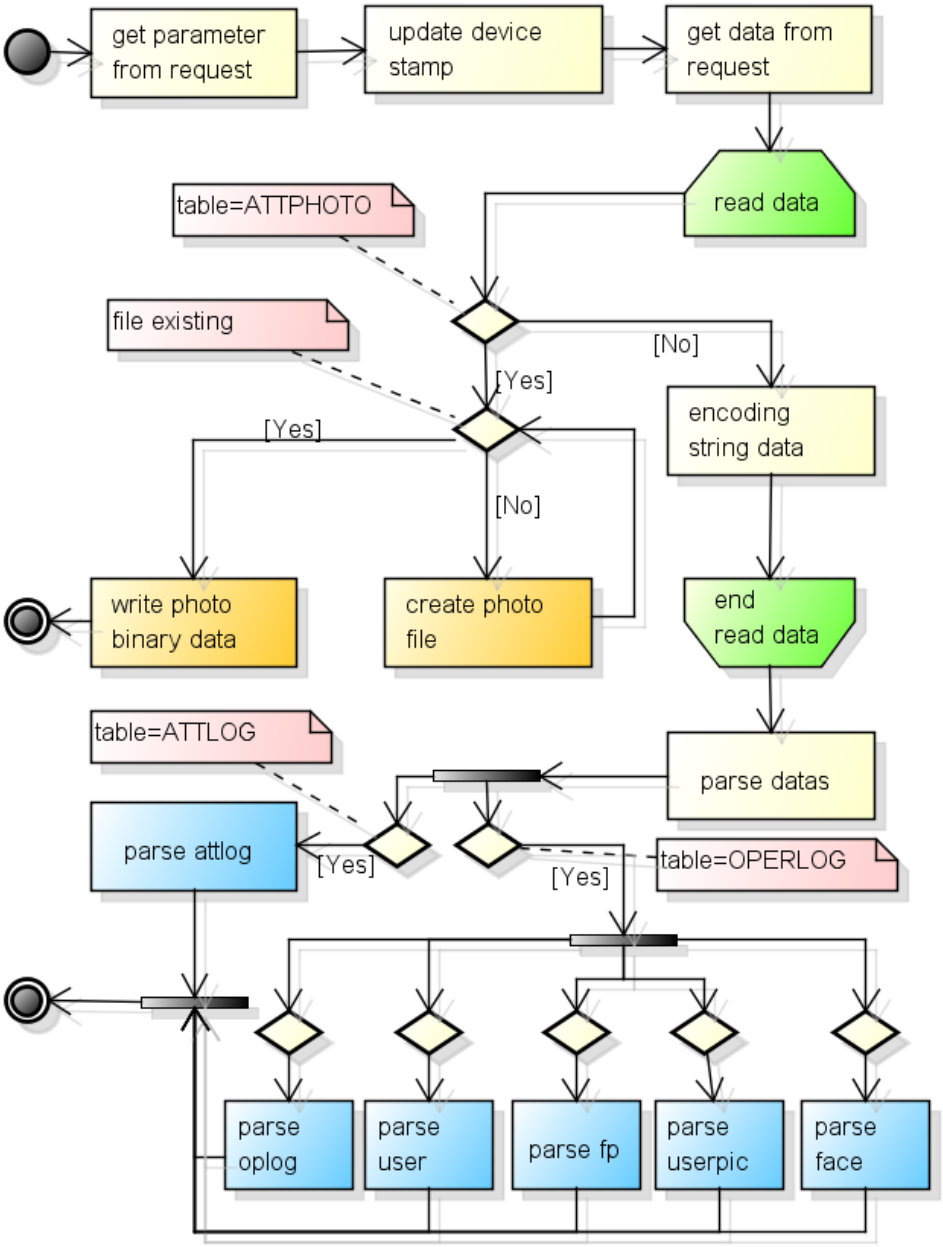
- Exchanging initialization information.
- Registering attendance remotely.



6.1.2 POST cdata

The operations that use the POST cdata method in server requests include:

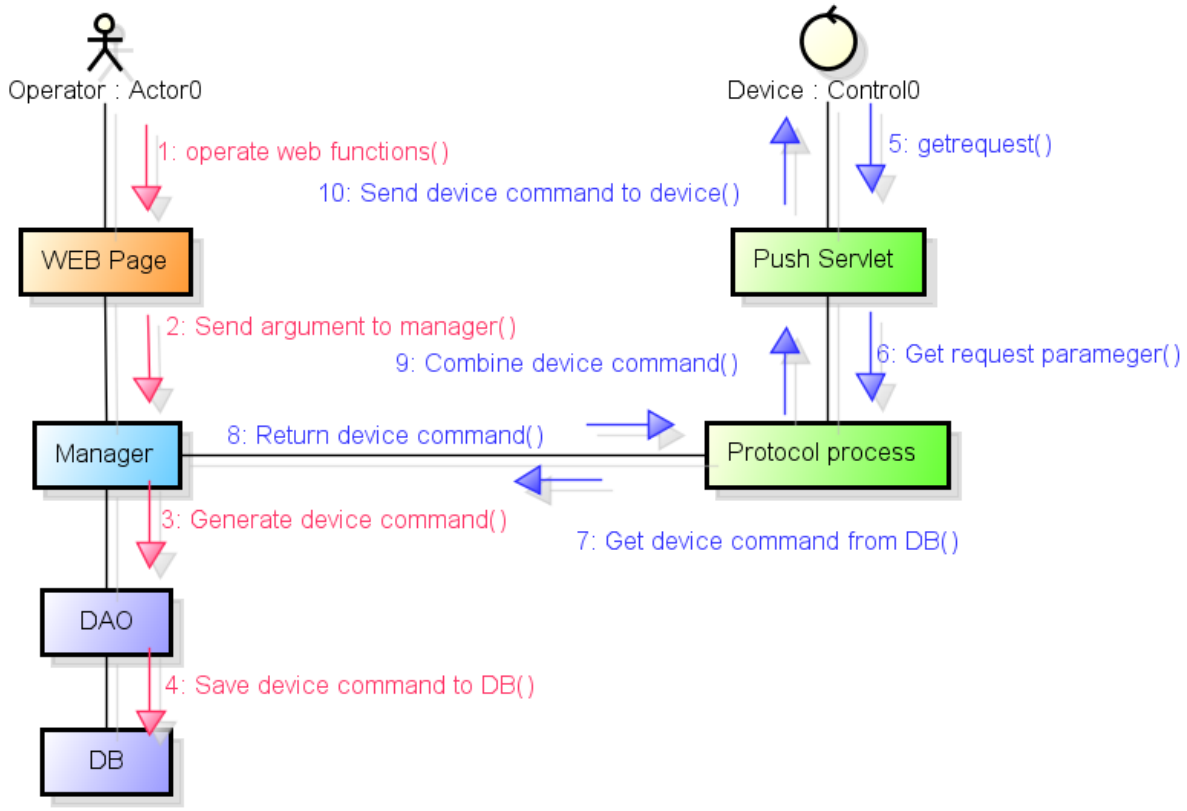
- Uploading user data.
- Uploading attendance logs.
- Uploading attendance photos.
- Uploading operation logs.
- Uploading the fingerprint data template.
- Uploading the face data template.
- Uploading user photos.



6.1.3 getrequest

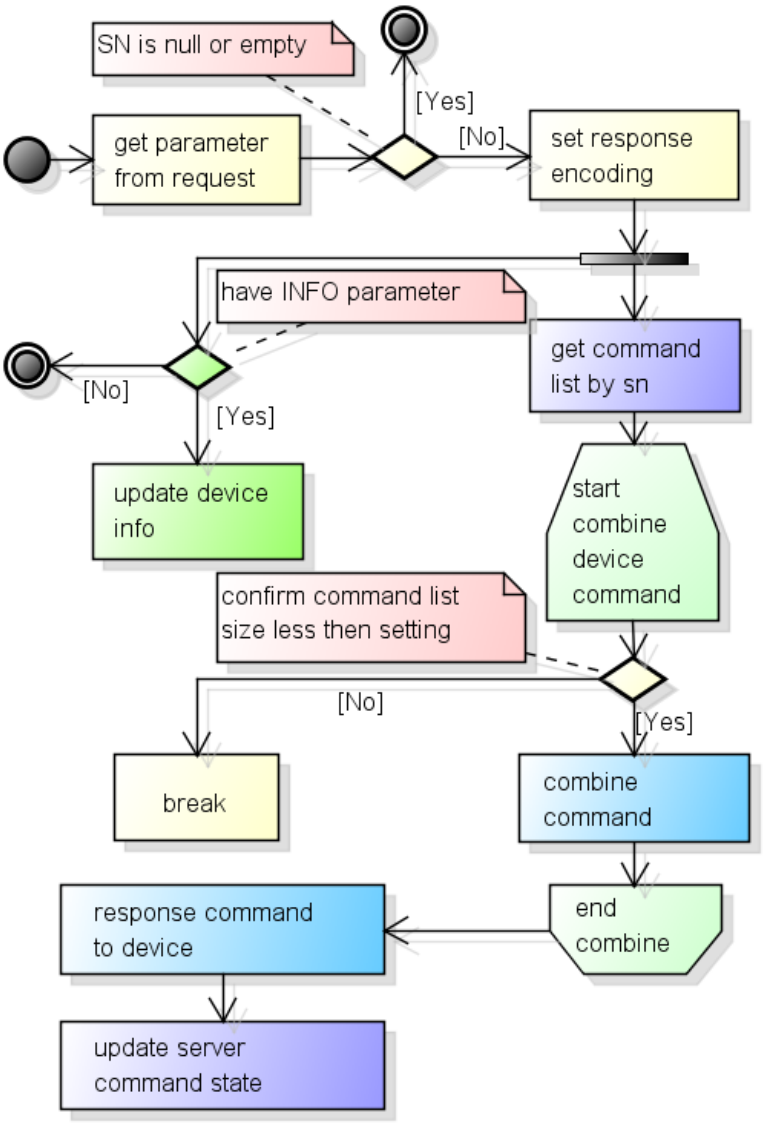
Device commands are generated by the server and are received and executed by devices.

- A customer generates a command in advance on the server based on service requirements, and saves the command to the database. For details, see Section 6.5 "Downloading Data from the Server to a Device" section. This step is illustrated by the text of the red arrows on the left in the following figure.
- A device regularly sends the getrequest request to the server to obtain the command. This step is illustrated by the text of the blue arrow on the right in the following figure.



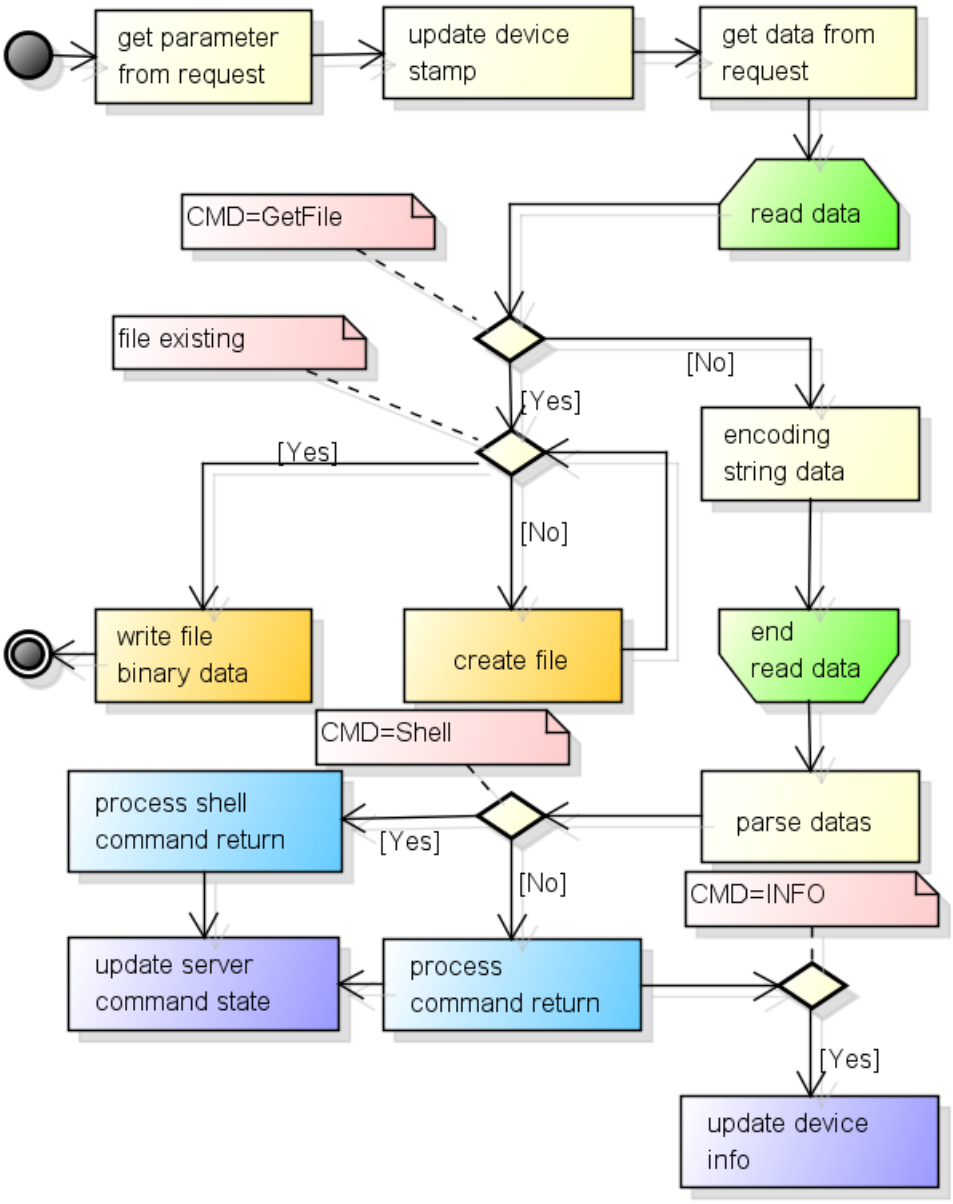
According to the protocol, the operations that use the getrequest method in server requests include:

- Uploading device update information.
- Obtaining a command for a device.



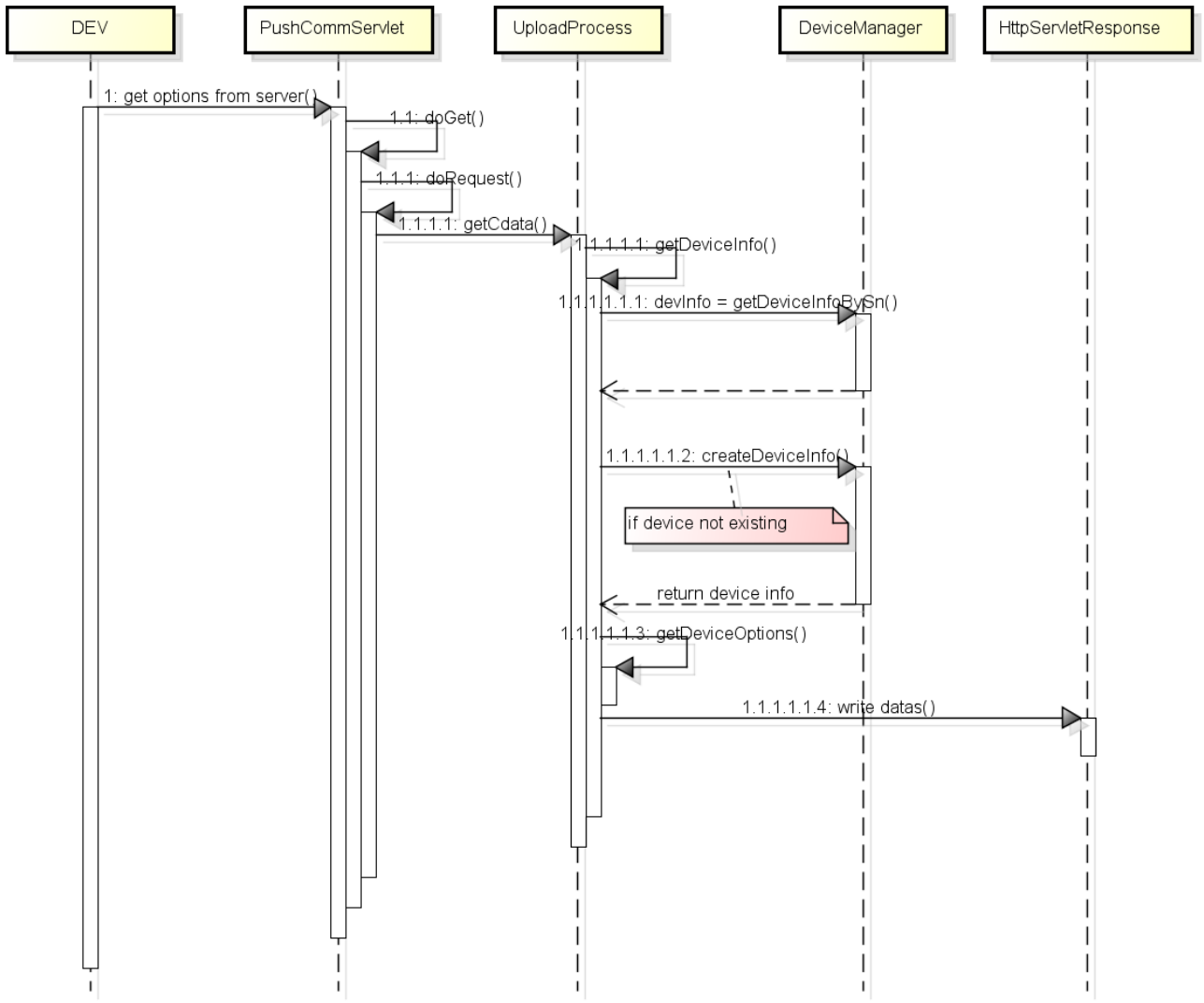
6.1.4 devicecmd

After a device command is executed, the devicecmd command is used to return the execution result to the server.

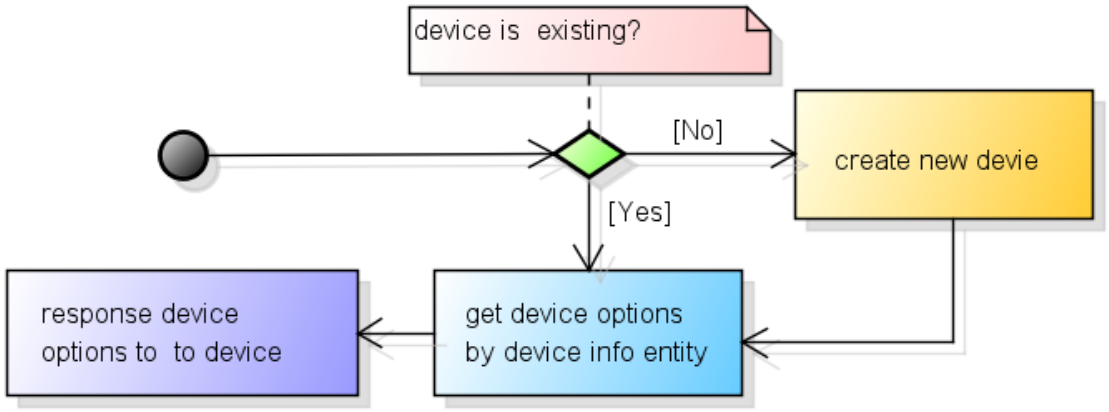


6.2 Exchanging Initialization Information

- For details of requests from devices, see the protocol document.
- The following figure shows the request sequence.

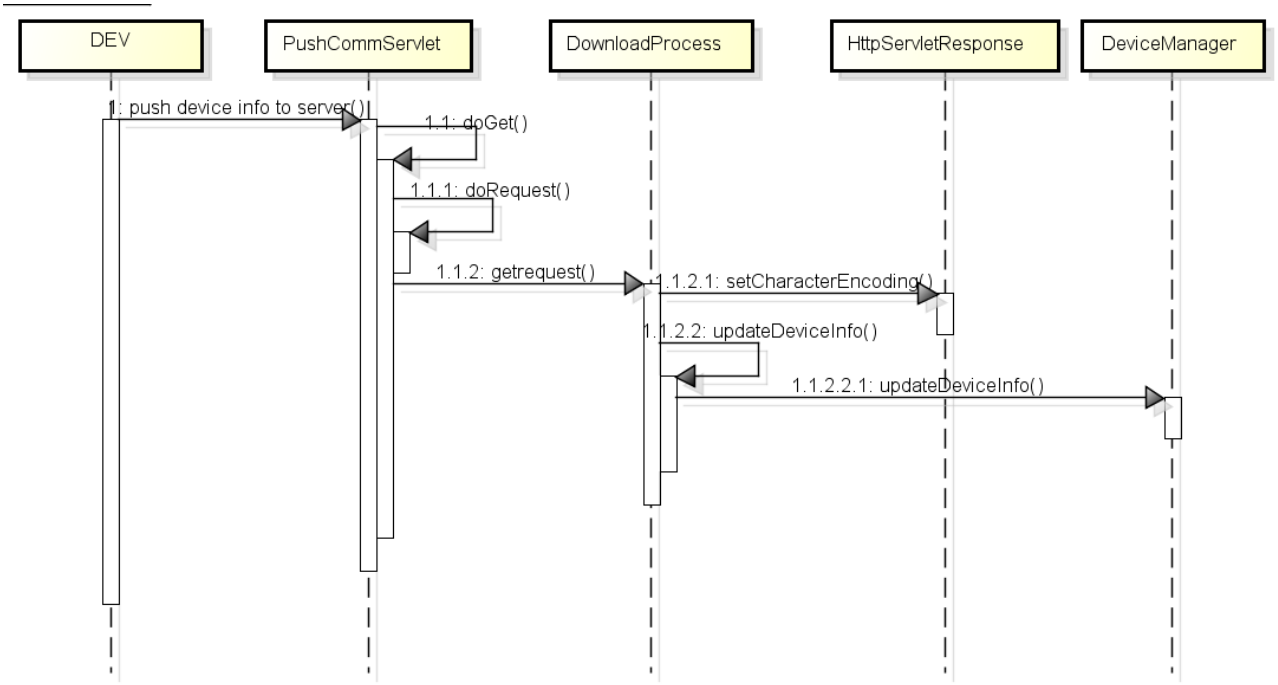


The following figure shows the request flowchart.



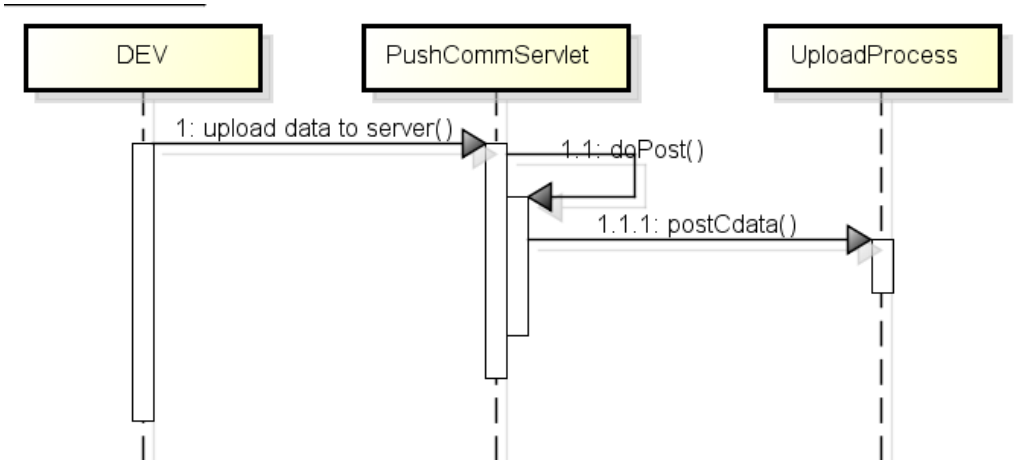
6.3 Uploading Device Update Information

- For details of requests from devices, see the protocol document.
- The following figure shows the request sequence.

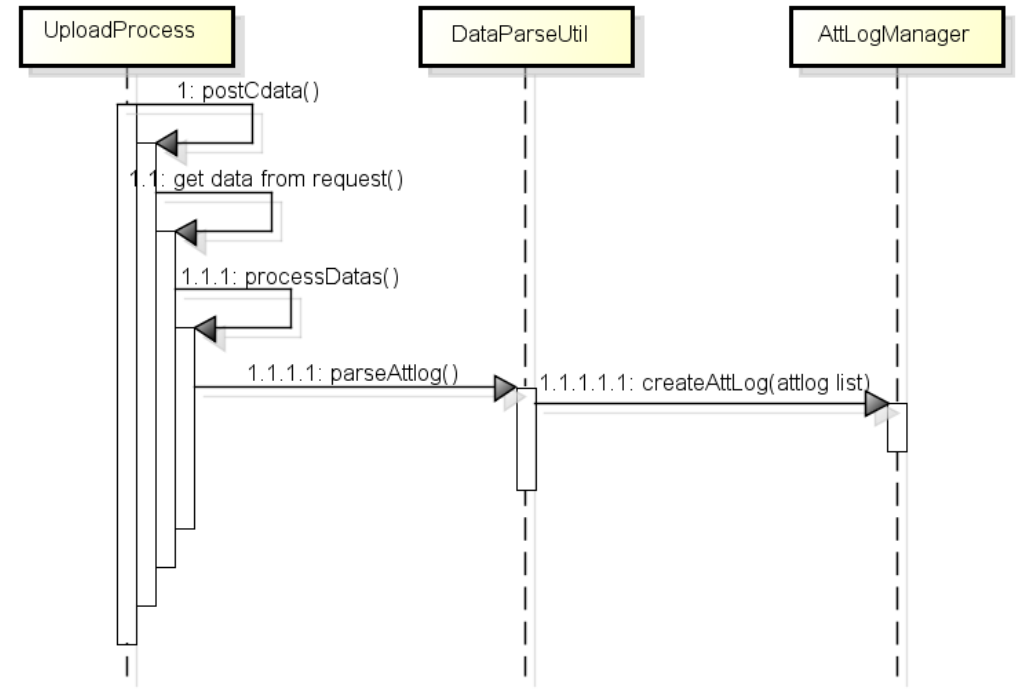


6.4 Uploading Device Data to the Server

Device data is uploaded using the Post method. The following figure shows the sequence.



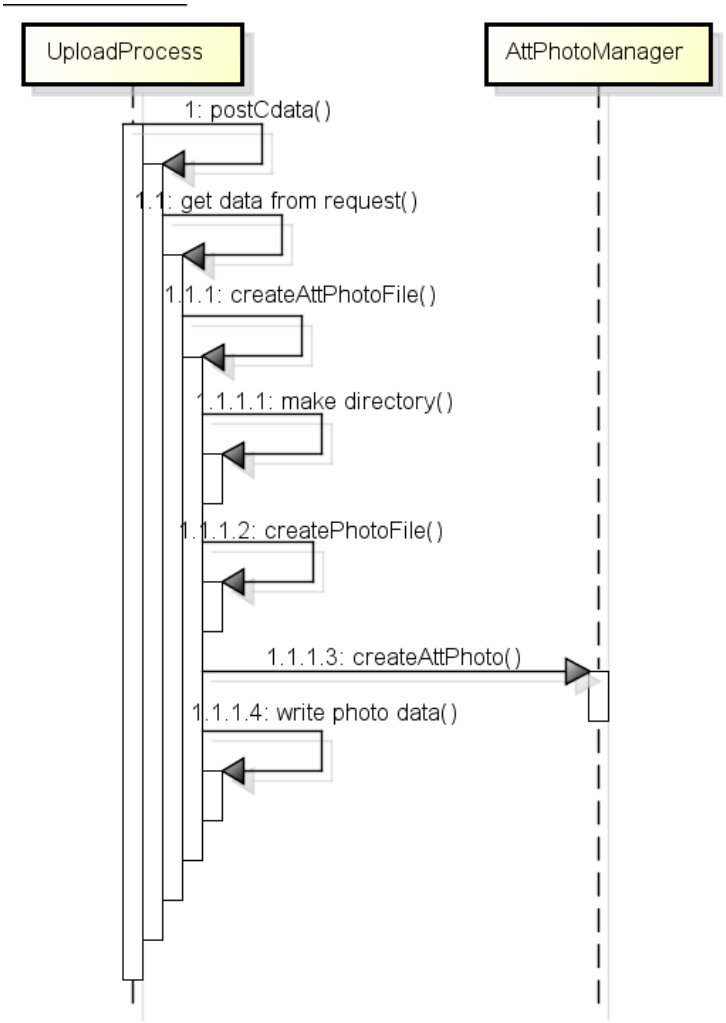
6.4.1 Uploading Attendance Logs



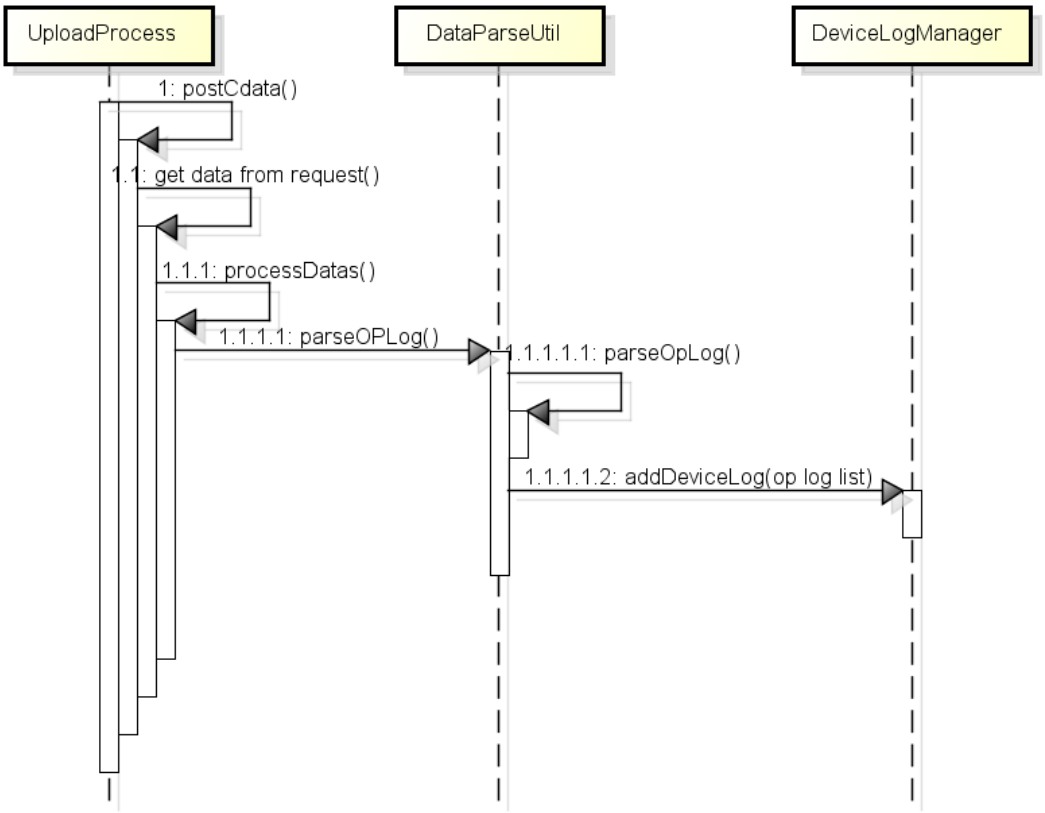
6.4.2 Uploading Attendance Photos

Attendance photos consist of attendance photo information and attendance photo data, and must be saved as photo files.

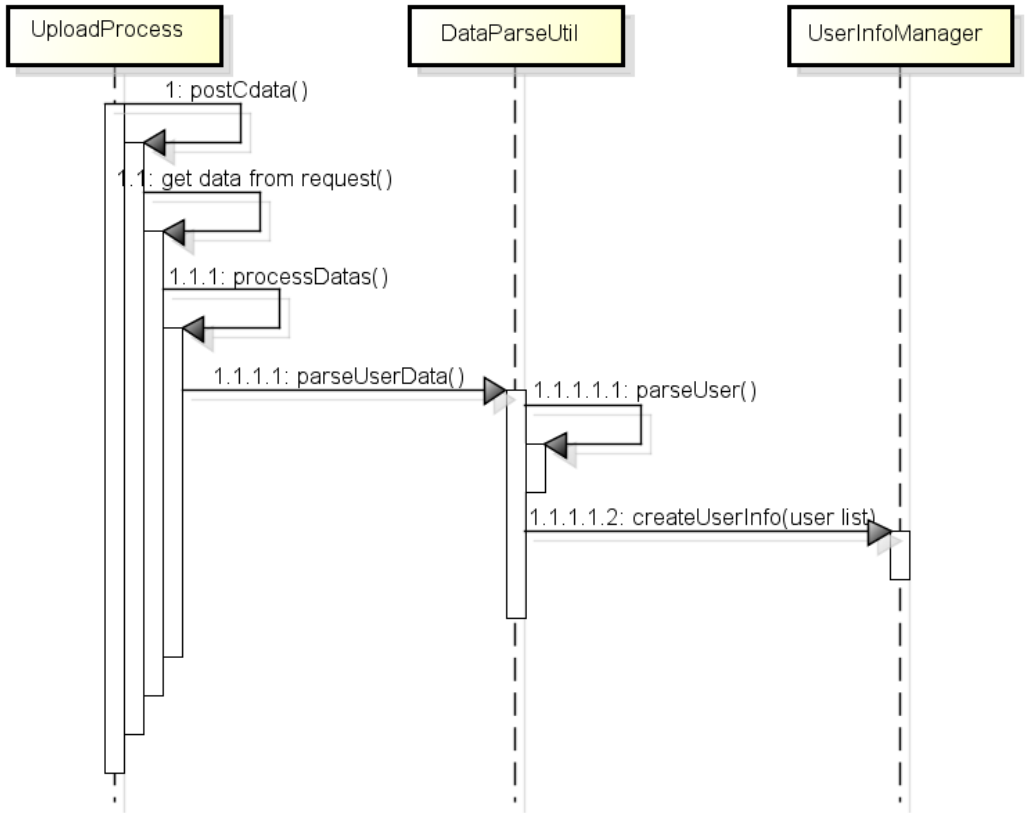
Sequence diagram:


UploadProcess
AttPhotoManager
1: postCdata()
1.1: get data from request()
1.1.1: createAttPhotoFile()
1.1.1.1: make directory()
1.1.1.2: createPhotoFile()
1.1.1.3: createAttPhoto()
1.1.1.4: write photo data()

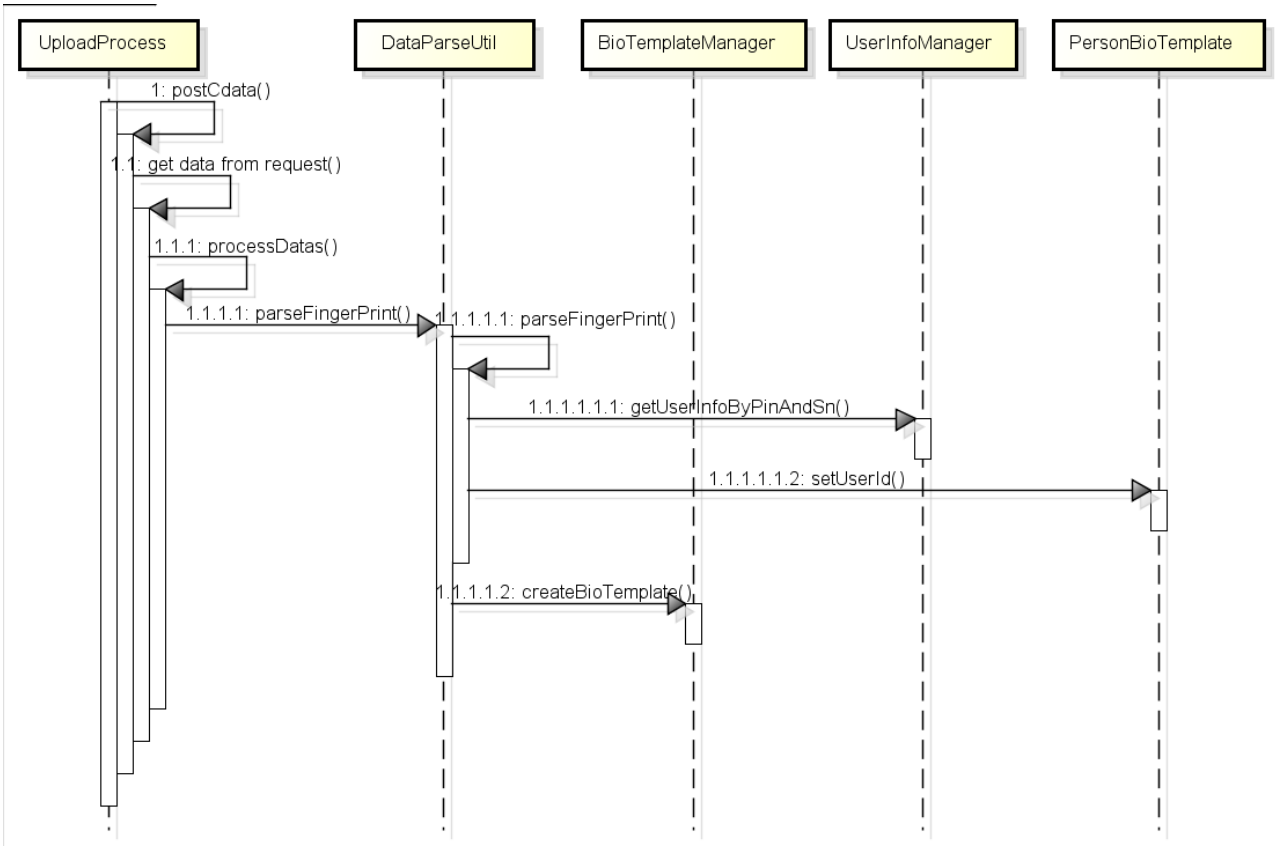
6.4.3 Uploading Operation Logs



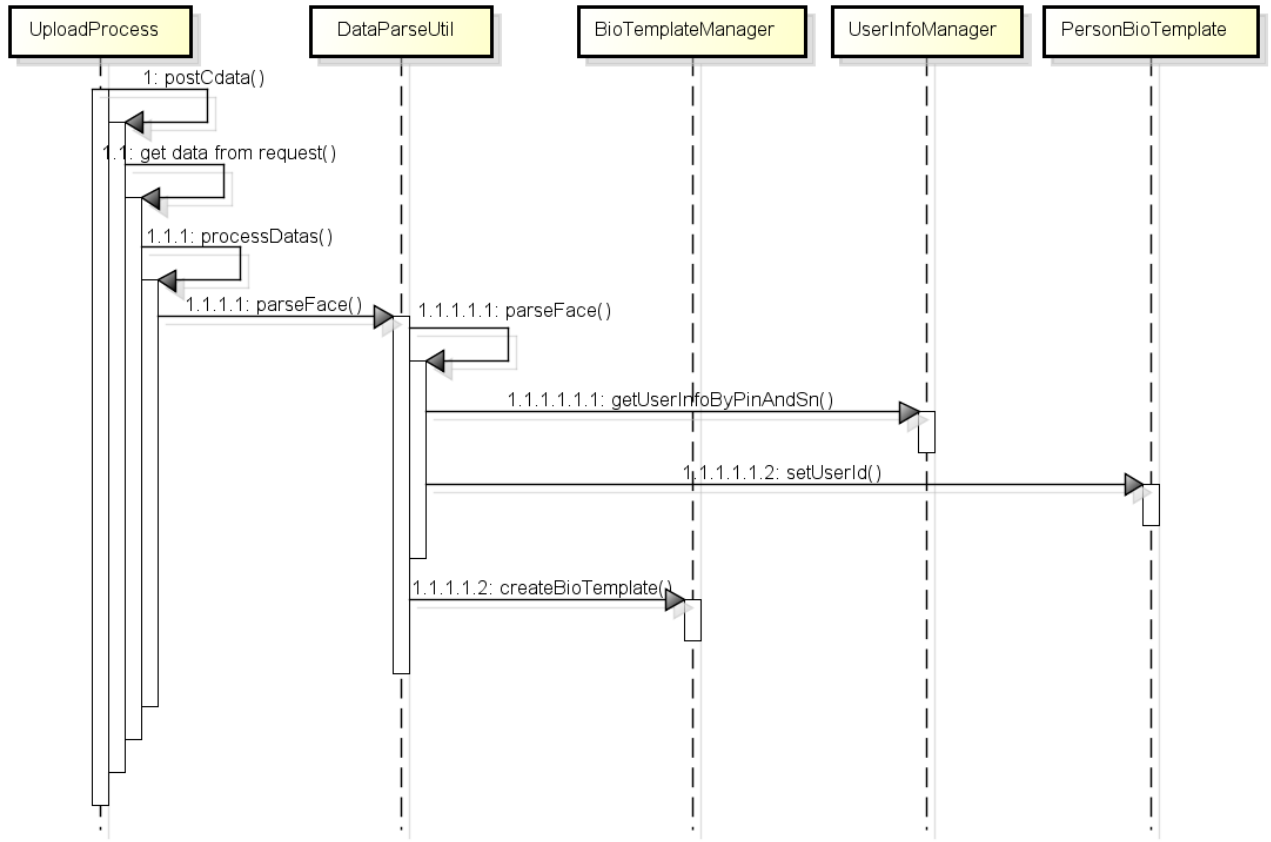
6.4.4 Uploading User Information



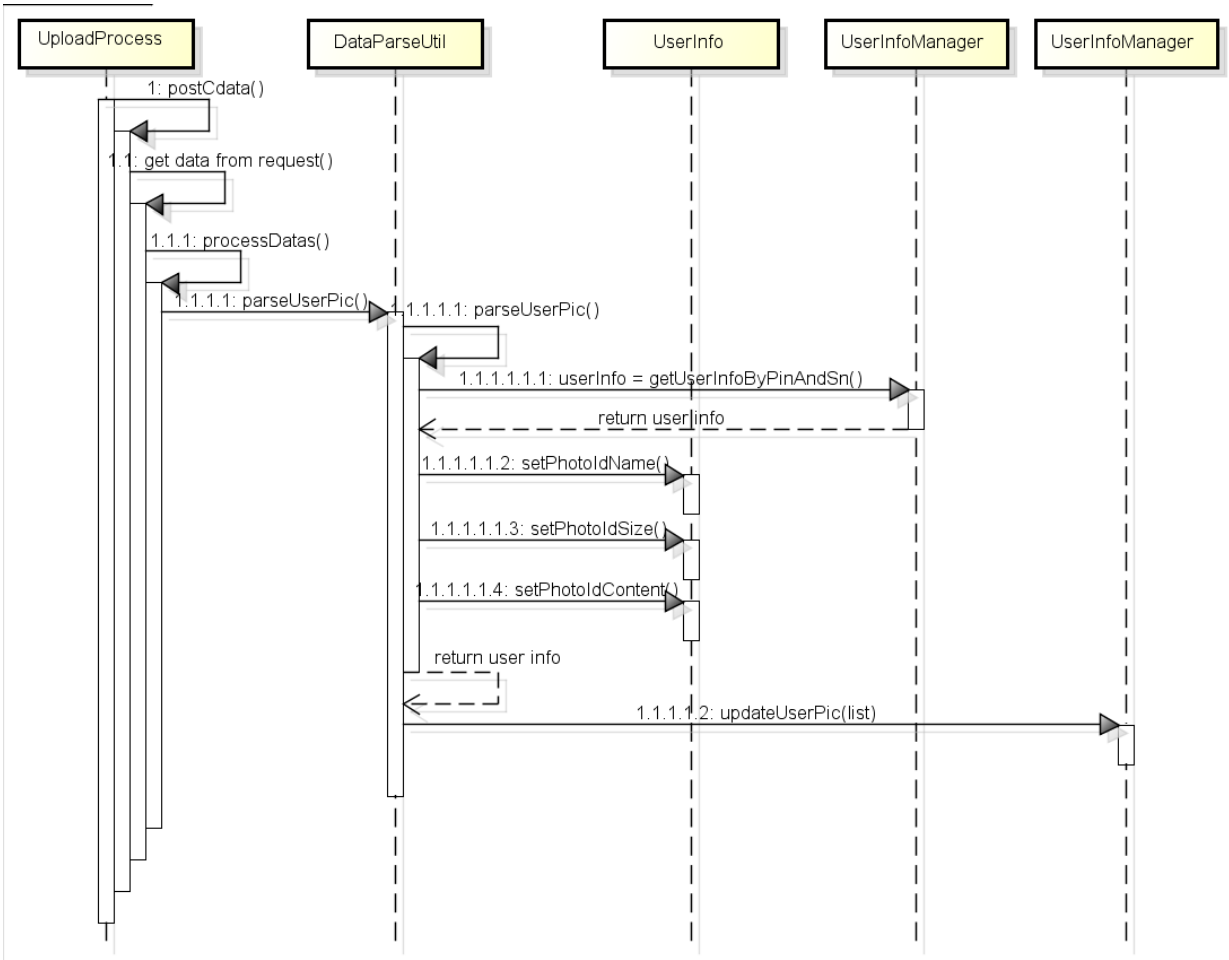
6.4.5 Uploading the Fingerprint Data Template



6.4.6 Uploading the Face Data Template

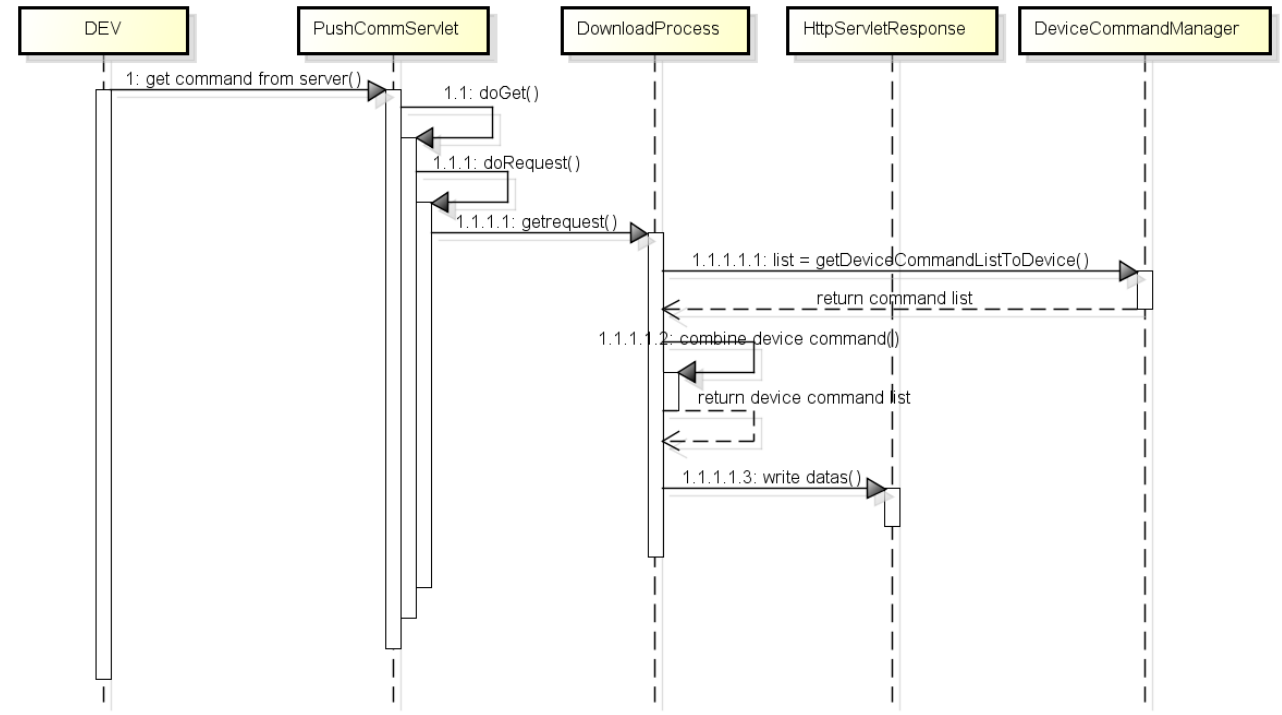


6.4.7 Uploading User Photos



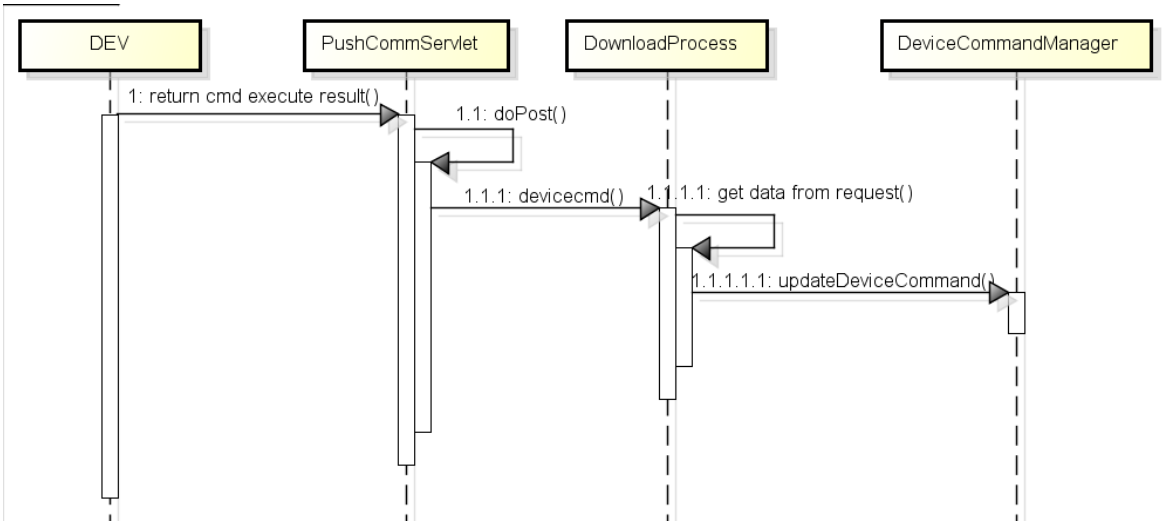
6.5 Downloading Data from the Server to a Device

6.5.1 Obtaining a Command for a Device



6.5.2 Returning a Command Execution Result

After obtaining a command from the server and executing it, a device must return the execution result to the server. The following figure shows the basic process of returning a command execution result. The process corresponding to a specific command is described in the related command delivery section.



6.5.3 DATA UPDATE Command - User Information

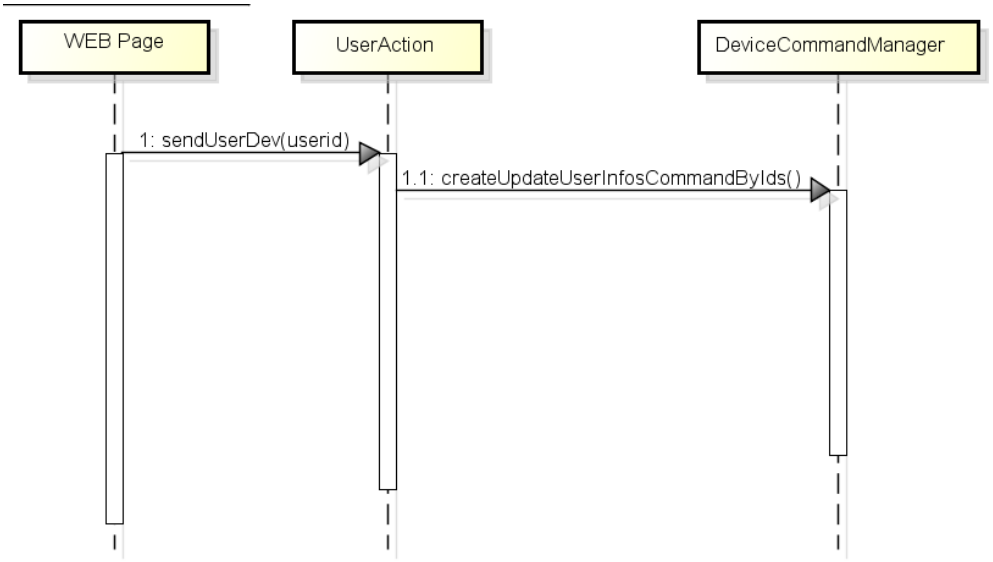
This command updates user information for devices. The operations involving this command in the Demo include:

- Device management.

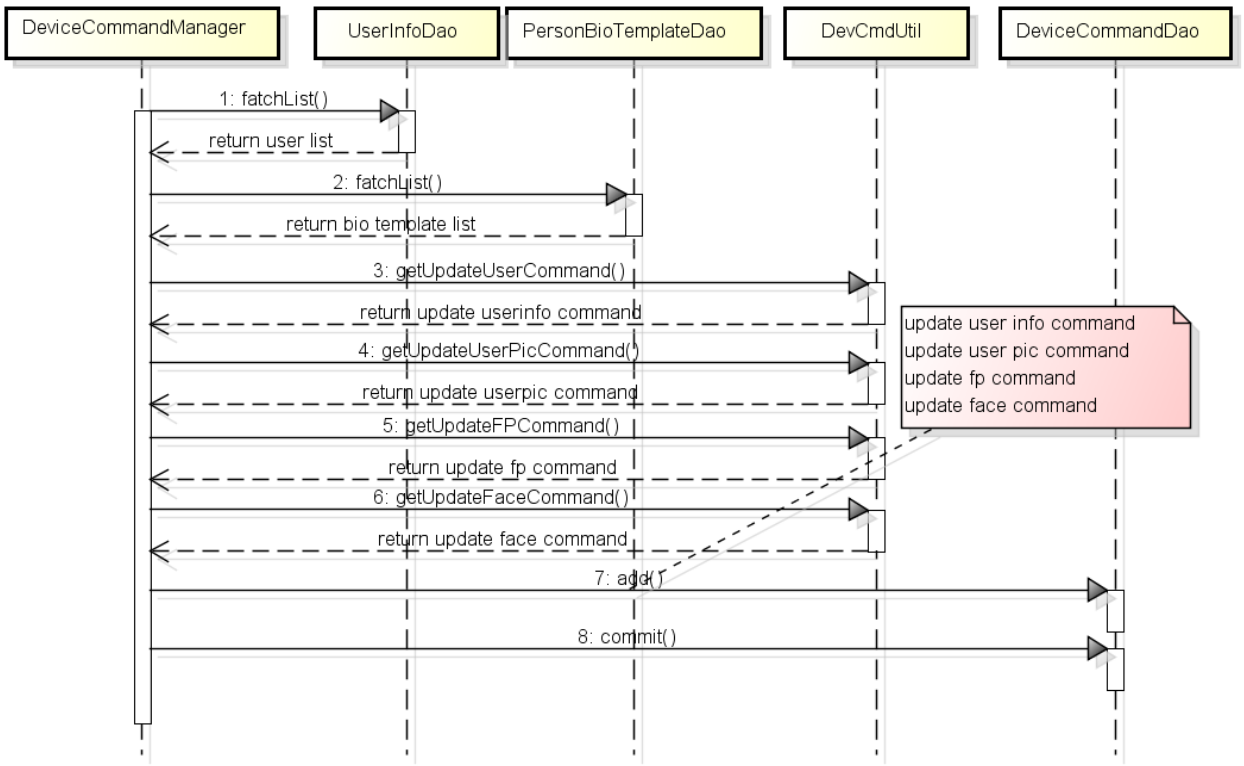
- Backing up data to other devices (using the DATA UPDATE command to back up user, fingerprint, face, and user photo data).
- Restoring data (using the DATA UPDATE command to restore the user, fingerprint, face, and user photo data).

- Personnel management
  - Sending personnel data to a device (using the DATA UPDATE command to send user, fingerprint, face, and user photo data).
  - Migrating personnel data to a new device (using the DATA UPDATE command to migrate user, fingerprint, face, and user photo data).

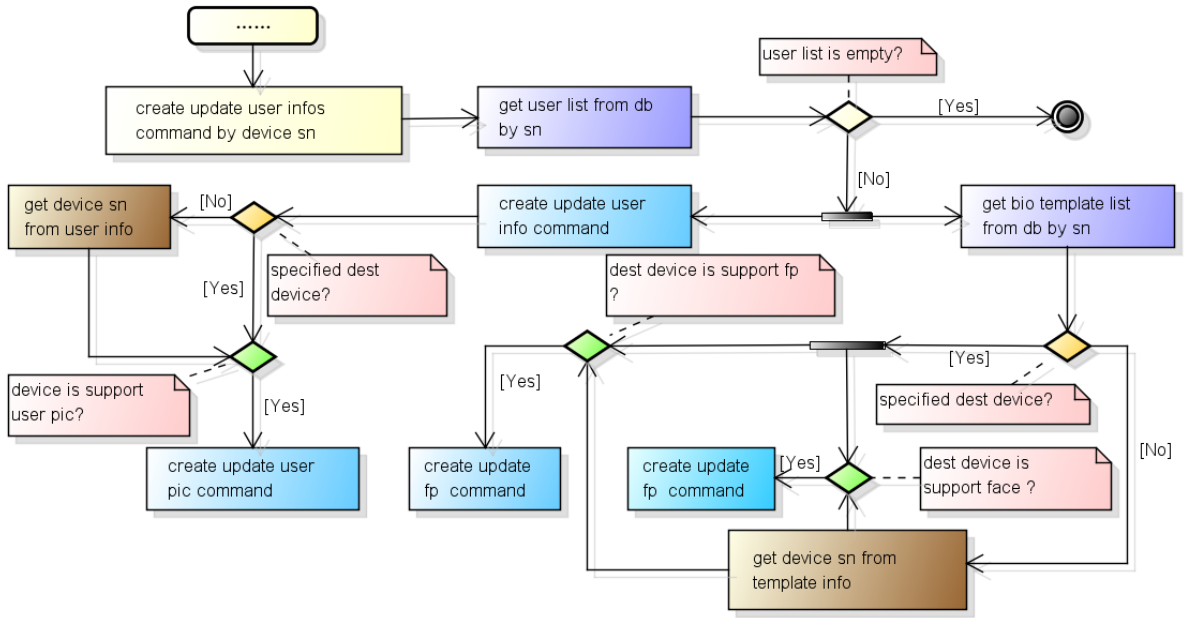
The sequence of the function for sending personnel data to a device is taken as an example. See the following figure.



Sequence diagram of the DeviceCommandManager$createUpdateUserInfosCommandByIds method:



Flowchart:



6.5.4 DATA UPDATE Command - Fingerprint Data Template

This command updates the fingerprint data template for devices. The operations involving this command in the Demo include:

- Device management
  - Backing up data to other devices (using the DATA UPDATE command to back up user, fingerprint, face, and user photo data).

- Restoring data (using the DATA UPDATE command to restore the user, fingerprint, face, and user photo data)

- **Personnel management**
  - Sending personnel data to a device (using the DATA UPDATE command to send user, fingerprint, face, and user photo data).
  - Migrating personnel data to a new device (using the DATA UPDATE command to migrate user, fingerprint, face, and user photo data).

See Section 6.5.3 "**DATA UPDATE Command - User Information**."

6.5.5 DATA UPDATE Command - Face Data Template

This command updates the face data template for devices. The operations involving this command in the Demo include:

- **Device management**
  - Backing up data to other devices (using the DATA UPDATE command to back up user, fingerprint, face, and user photo data).
  - Restoring data (using the DATA UPDATE command to restore the user, fingerprint, face, and user photo data).

- **Personnel management**
  - Sending personnel data to a device (using the DATA UPDATE command to send user, fingerprint, face, and user photo data).
  - Migrating personnel data to a new device (using the DATA UPDATE command to migrate user, fingerprint, face, and user photo data).

See Section 6.5.3 "DATA UPDATE Command - User Information."

6.5.6 DATA UPDATE Command - User Photos

This command updates user photos for devices. The operations involving this command in the Demo include:

- **Device management**

Backing up data to other devices (using the DATA UPDATE command to back up user, fingerprint, face, and user photo data).

Restoring data (using the DATA UPDATE command to restore the user, fingerprint, face, and user photo data).

- **Personnel management**

Sending personnel data to a device (using the DATA UPDATE command to send user, fingerprint, face, and user photo data).

Migrating personnel data to a new device (using the DATA UPDATE command to migrate user, fingerprint, face, and user photo data).

See Section 6.5.3 "DATA UPDATE Command - User Information."

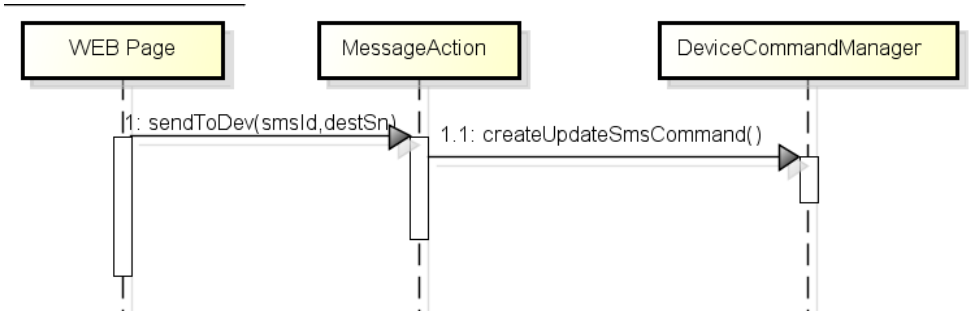
6.5.7 DATA UPDATE Command - SMS Messages

This command updates SMS messages for devices. The operations involving this command in the Demo include:

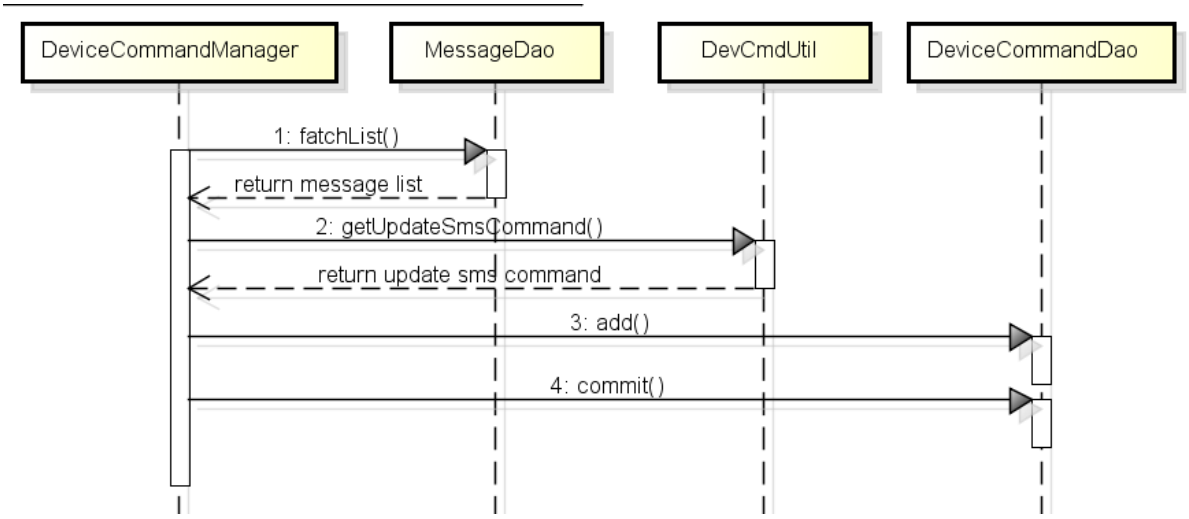
- SMS messages

Sending SMS messages to devices (using the DATA UPDATE SMS command).

Sequence diagram:



Sequence diagram of the DeviceCommandManager$createUpdateSmsCommand method:



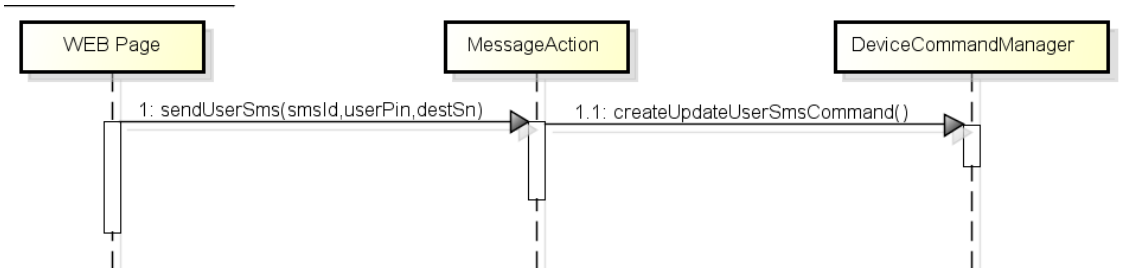
6.5.8 DATA UPDATE Command - Personal SMS Messages

This command updates personal SMS messages for devices. The operations involving this command in the Demo include:

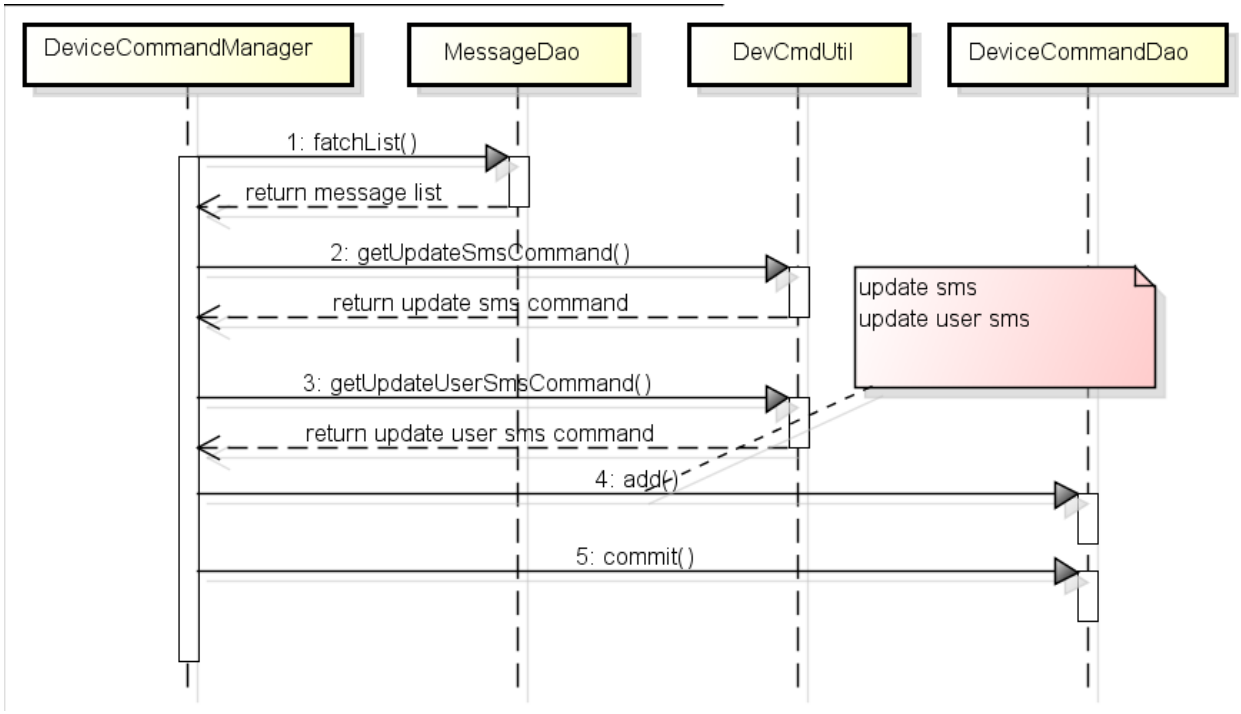
- SMS messages

Sending personal SMS messages to devices (using the DATA UPDATE USER_SMS command).

Sequence diagram:



Sequence diagram of the DeviceCommandManager$createUpdateUserSmsCommand method:



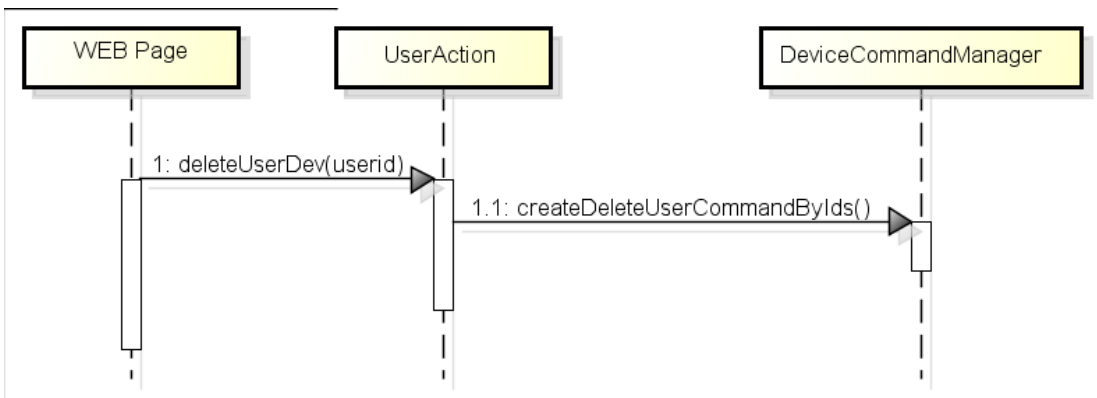
6.5.9 DATA DELETE Command - User Information

This command deletes user information from devices. The operations involving this command in the Demo include:

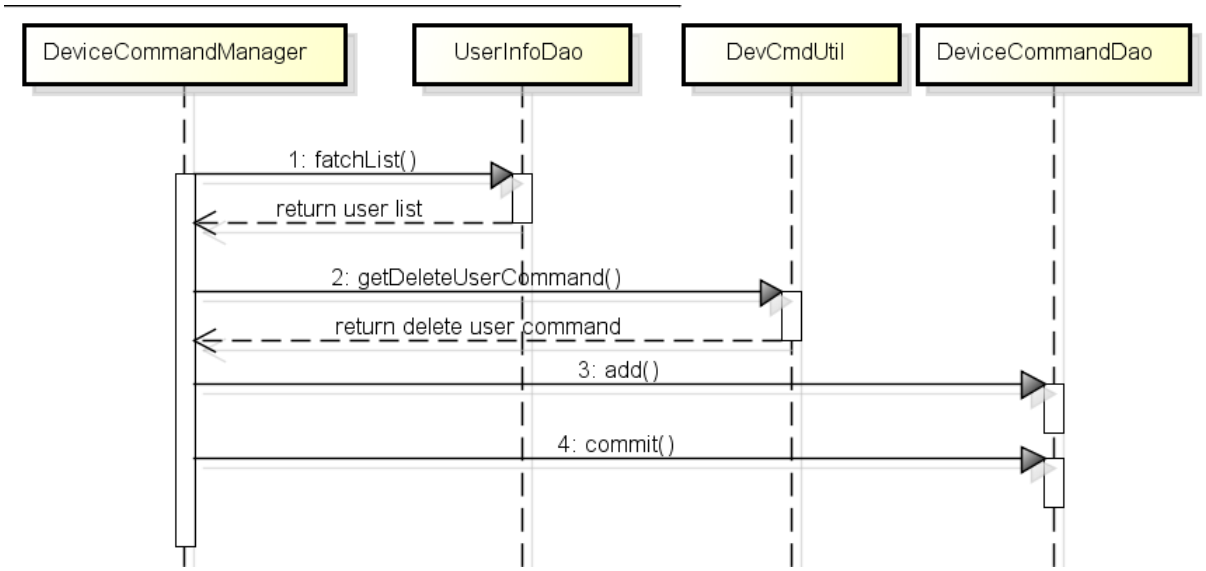
- Personnel management

Deleting personnel data from a device (using the DATA DELETE command to delete user, fingerprint, face, and user photo data).

Sequence diagram:



Sequence diagram of the DeviceCommandManager$createDeleteUserCommandByIds method:



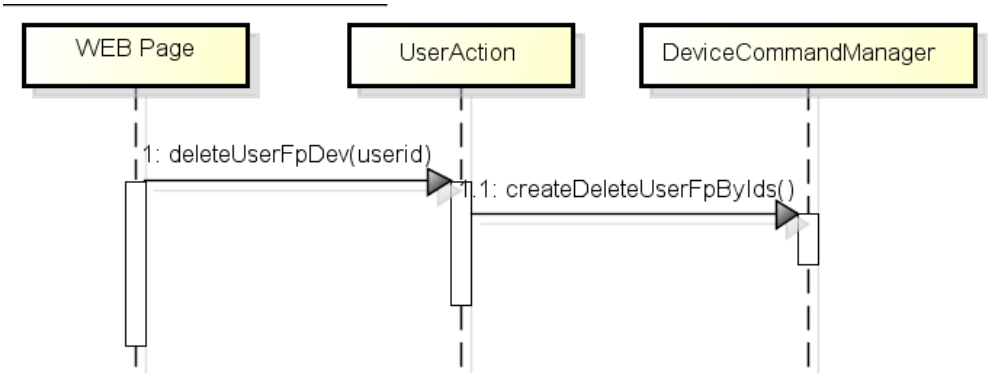
6.5.10 DATA DELETE Command - Fingerprint Data Template

This command deletes the fingerprint data template from devices. The operations involving this command in the Demo include:

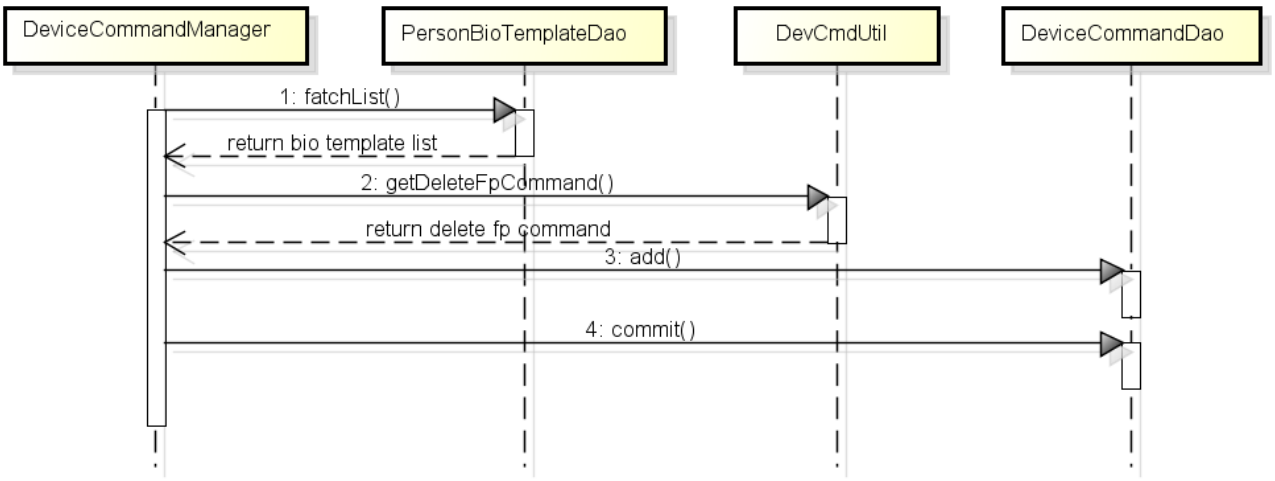
- Personnel management

Deleting the fingerprint data template from a device (using the DATA DELETE FINGERTMP command).

Sequence diagram:



Sequence diagram of the DeviceCommandManager$createDeleteUserFpByIds method:



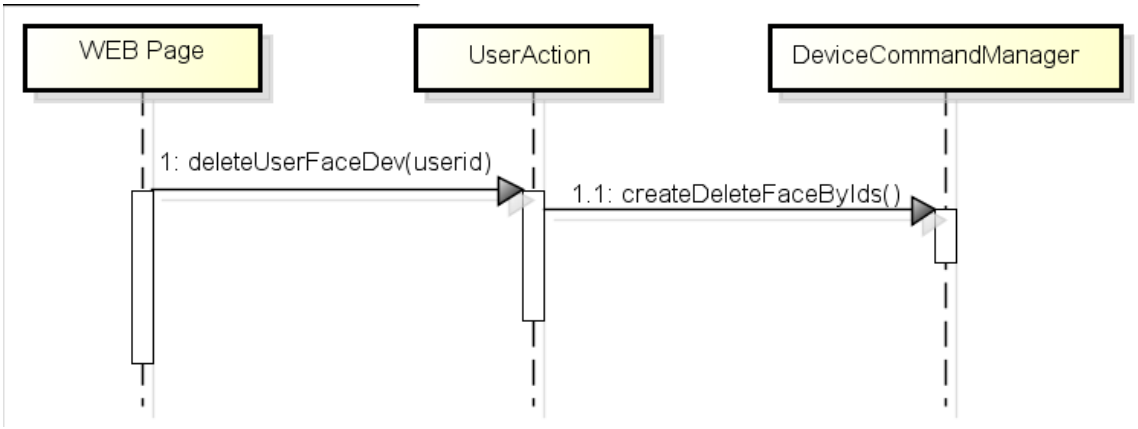
6.5.11 DATA DELETE Command - Face Data Template

This command deletes the face data template from devices. The operations involving this command in the Demo include:

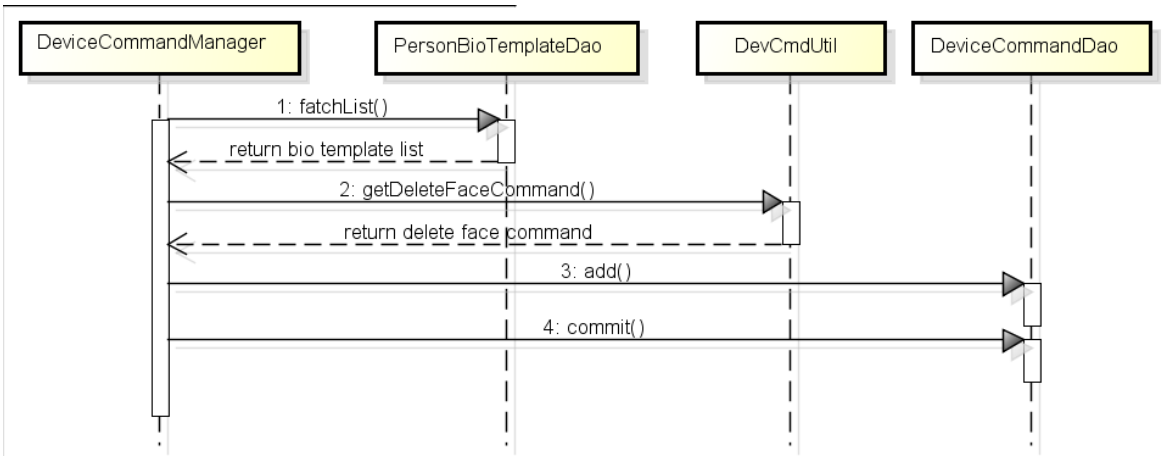
- Personnel management

Deleting the face data template from a device (using the DATA DELETE FACE command).

Sequence diagram:



Sequence diagram of the DeviceCommandManager$createDeleteFaceByIds method:



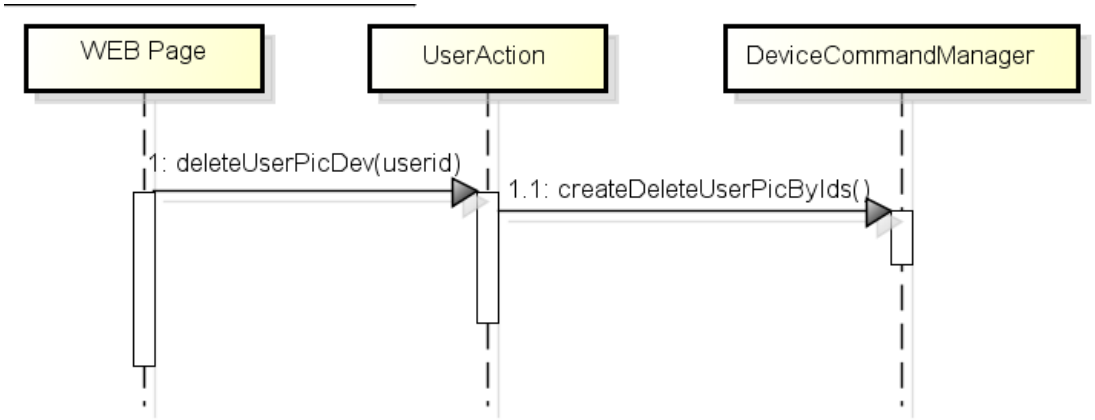
6.5.12 DATA DELETE Command - User Photos

This command deletes user photos from devices. The operations involving this command in the Demo include:

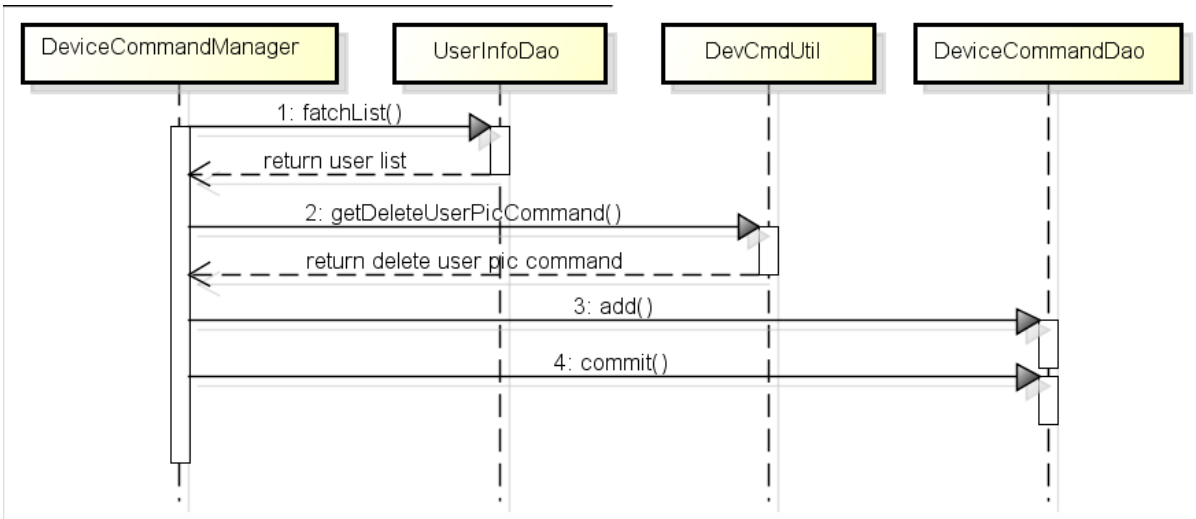
- Personnel management

Deleting user photos from a device (using the DATA DELETE USERPIC command).

Sequence diagram:



Sequence diagram of the DeviceCommandManager$createDeleteUserPicByIds method:



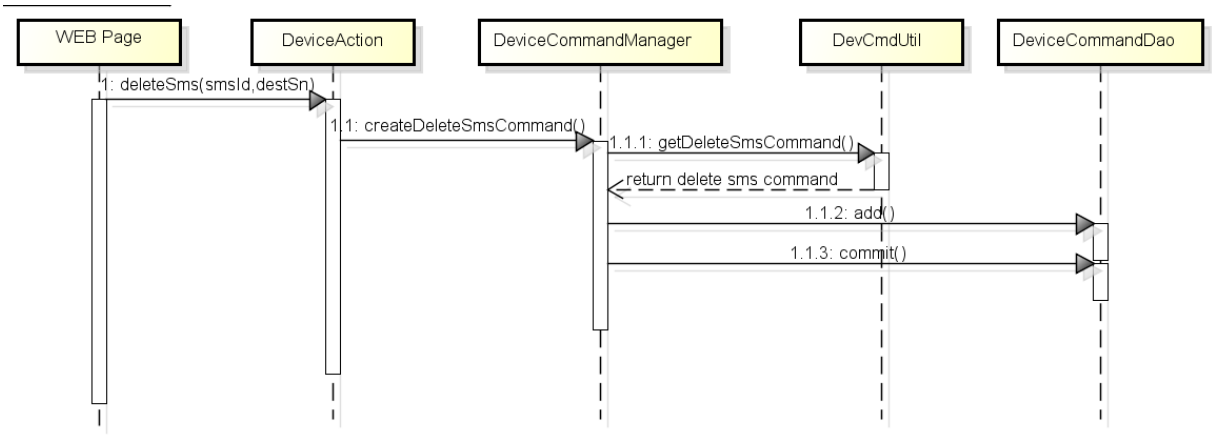
6.5.13 DATA DELETE Command - SMS Messages

This command deletes SMS messages for devices. The operations involving this command in the Demo include:

- Device management

Deleting SMS messages from a device (using the DATA DELETE SMS command).

Sequence diagram:



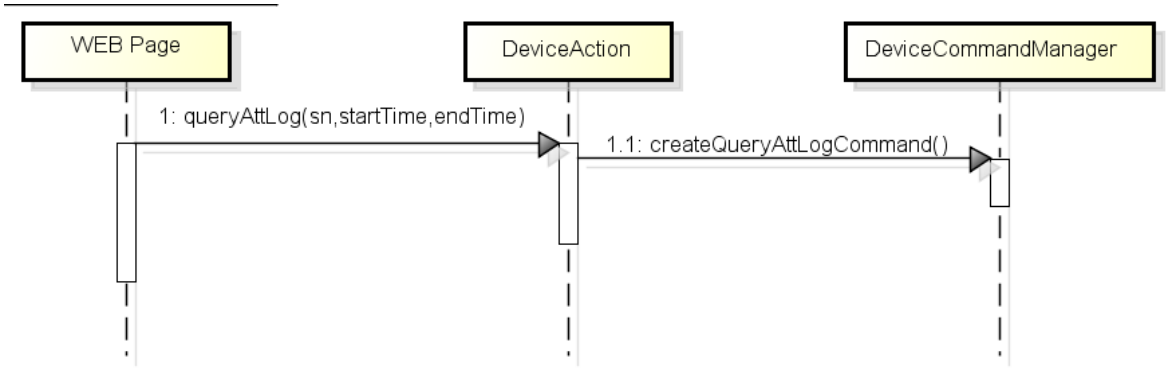
6.5.14 DATA QUERY Command - Attendance Logs

This command queries attendance logs for devices. The operations involving this command in the Demo include:

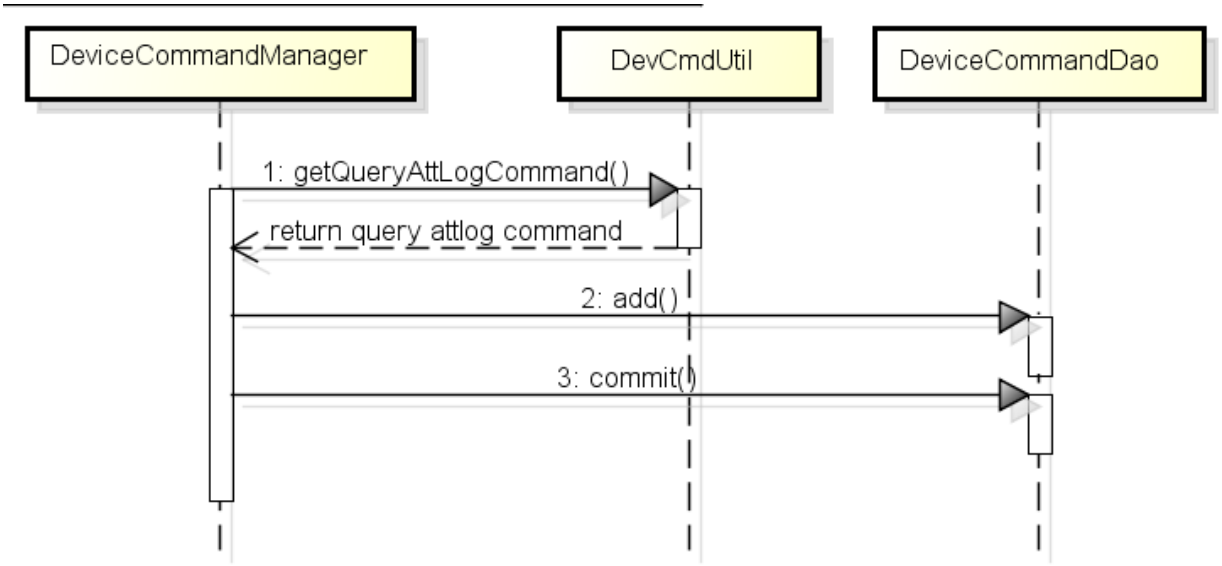
- Device management

Querying attendance logs (using the DATA QUERY ATTLOG command).

Sequence diagram:



Sequence diagram of the DeviceCommandManager$createQueryAttLogCommand method:



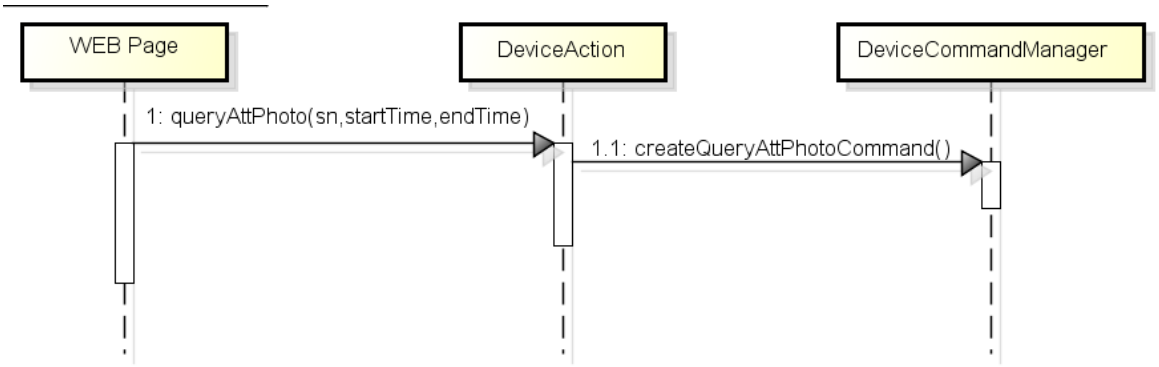
6.5.15 DATA QUERY Command - Attendance Photos

This command queries attendance photos for devices. The operations involving this command in the Demo include:

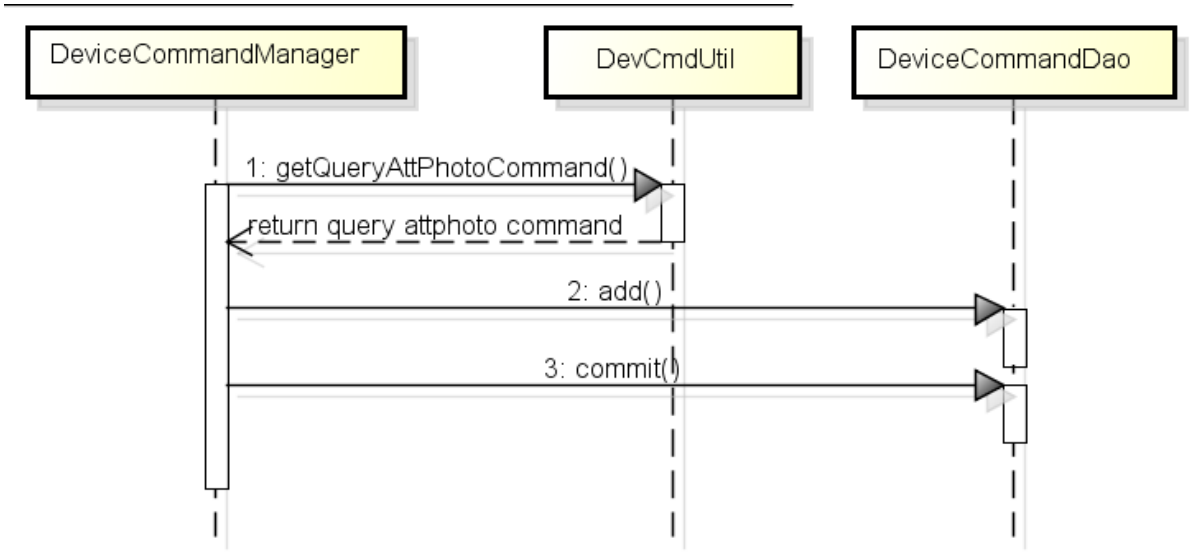
- **Device management**

Querying attendance photos (using the DATA QUERY ATTPHOTO command).

Sequence diagram:



Sequence diagram of the DeviceCommandManager$createQueryAttPhotoCommand method:



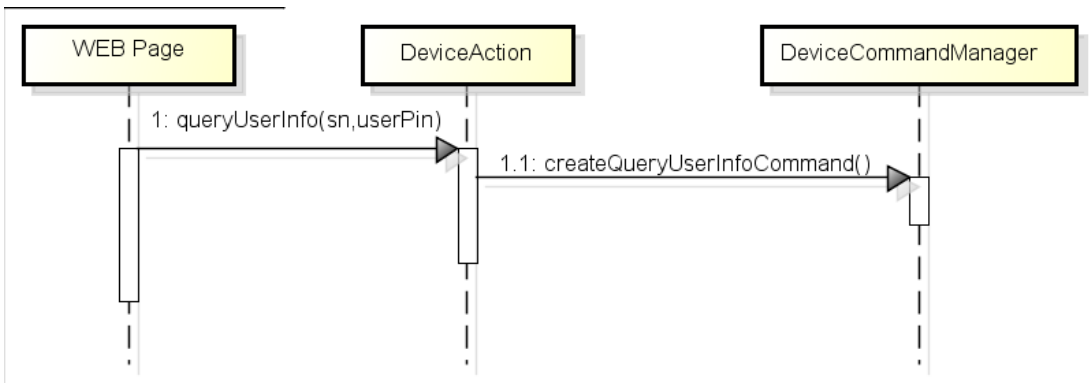
6.5.16 DATA QUERY Command - User Information

This command queries user information from devices. The operations involving this command in the Demo include:

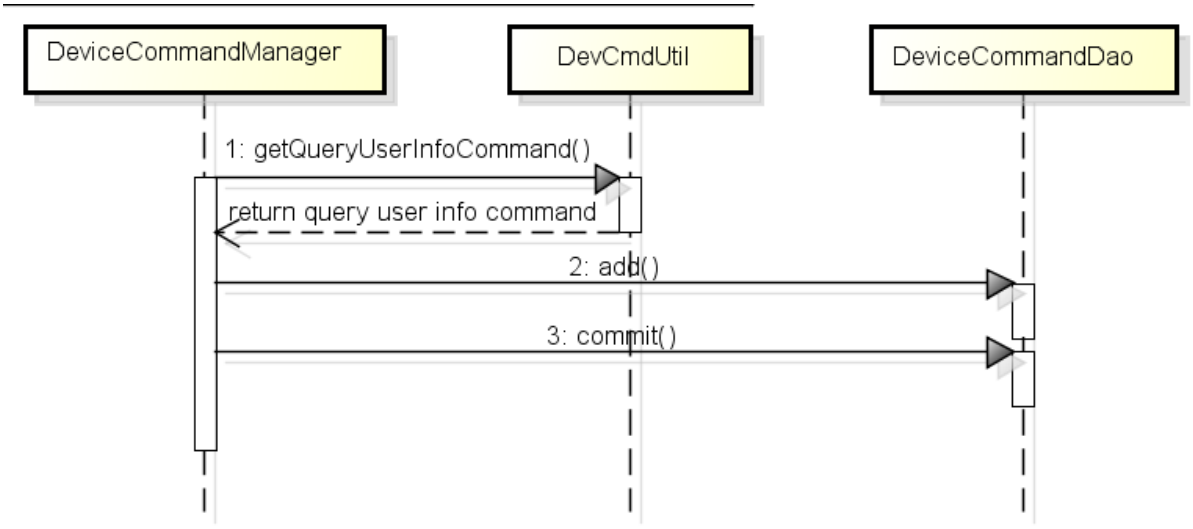
- **Device management**

Querying user information (using the DATA QUERY USERINFO command).

Sequence diagram:



Sequence diagram of the DeviceCommandManager$createQueryUserInfoCommand method:



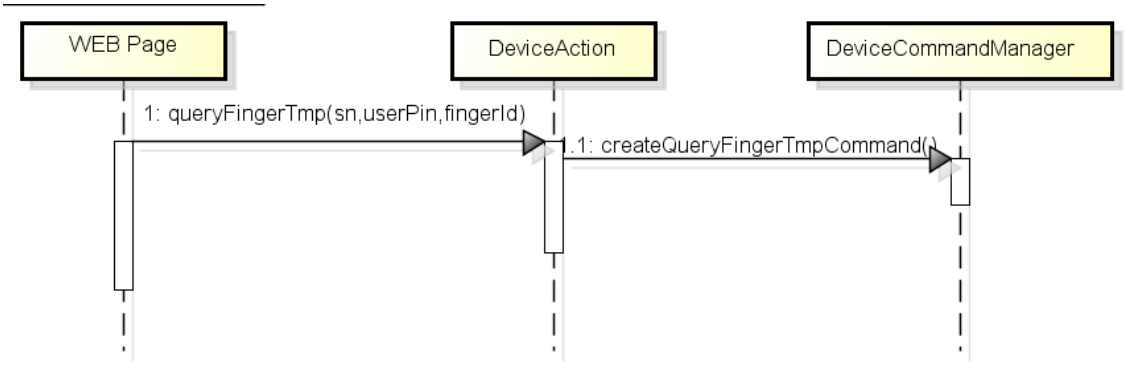
6.5.17 DATA QUERY Command - Fingerprint Data Template

This command queries the fingerprint data template for devices. The operations involving this command in the Demo include:

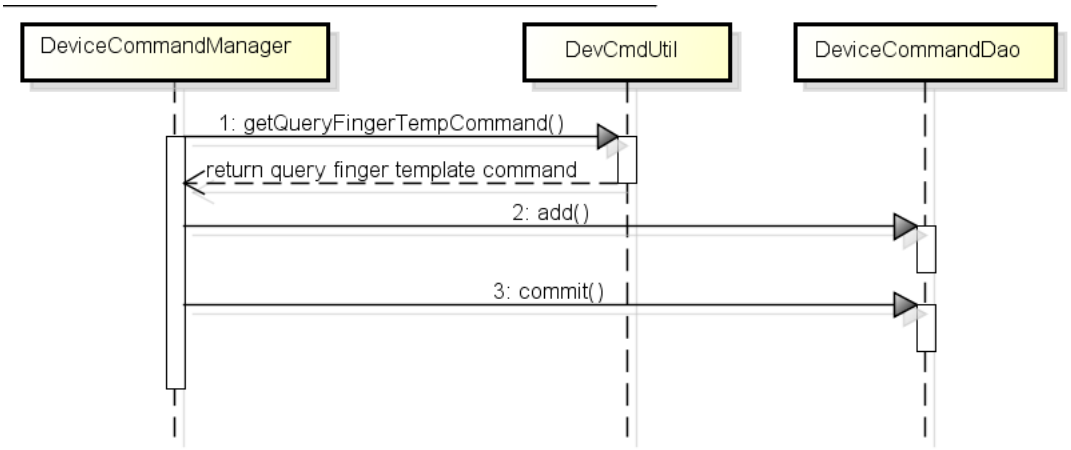
- **Device management**

Querying the fingerprint data template (using the DATA QUERY FINGERTMP command).

Sequence diagram:



Sequence diagram of the DeviceCommandManager$createQueryFingerTmpCommand method:



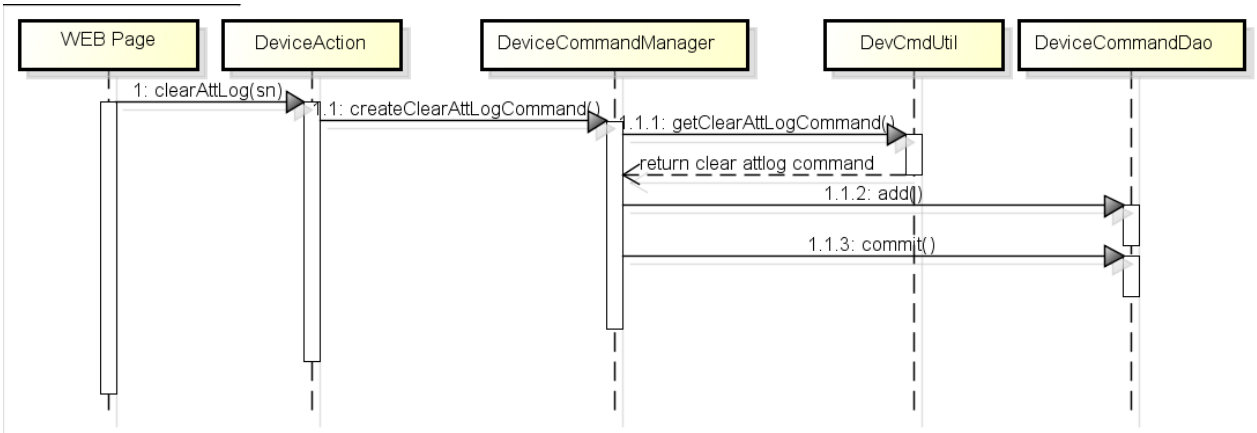
6.5.18 CLEAR Command - Attendance Logs

This command clears attendance logs for devices. The operations involving this command in the Demo include:

- **Device management**

Clearing attendance logs (using the CLEAR LOG command).

Sequence diagram:



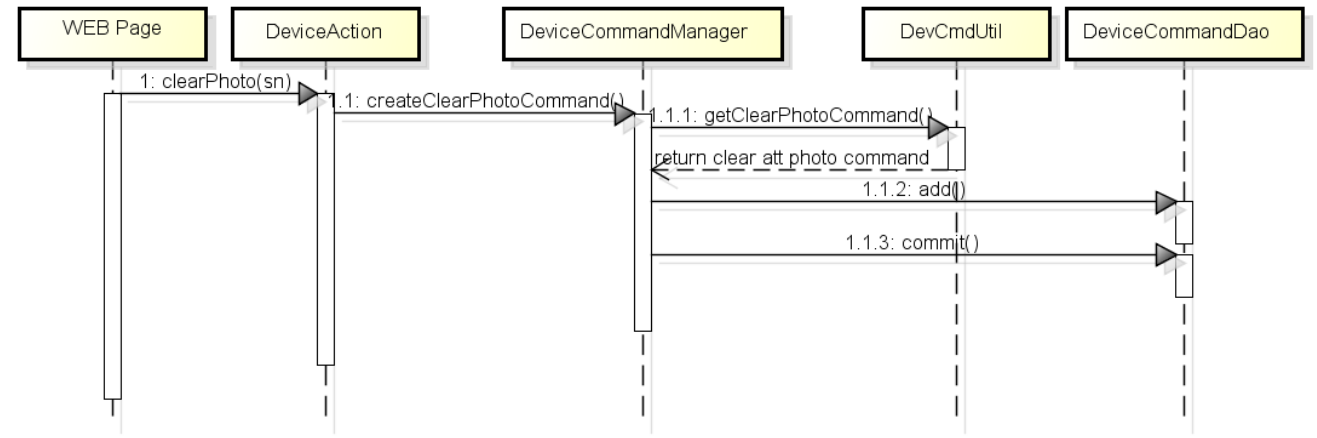
6.5.19 CLEAR Command - Attendance Photos

This command clears attendance photos for devices. The operations involving this command in the Demo include:

- **Device management**

Clearing attendance photos (using the CLEAR PHOTO command).

Sequence diagram:



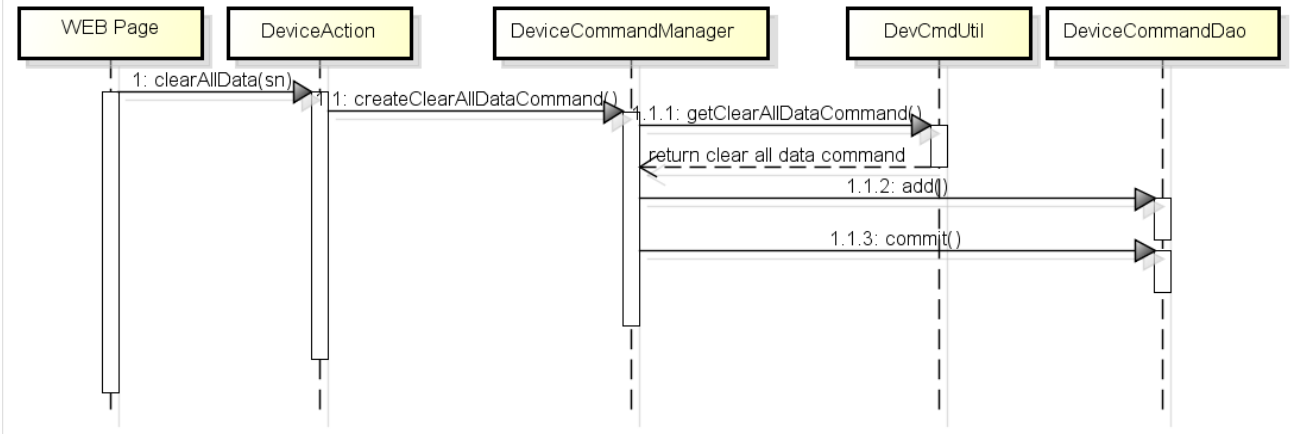
6.5.20 CLEAR Command - All Data

This command clears all data for devices. The operations involving this command in the Demo include:

- **Device management**

Clearing all data (using the CLEAR DATA command).

Sequence diagram:



6.5.21 Check Command - Checking for Data Updates

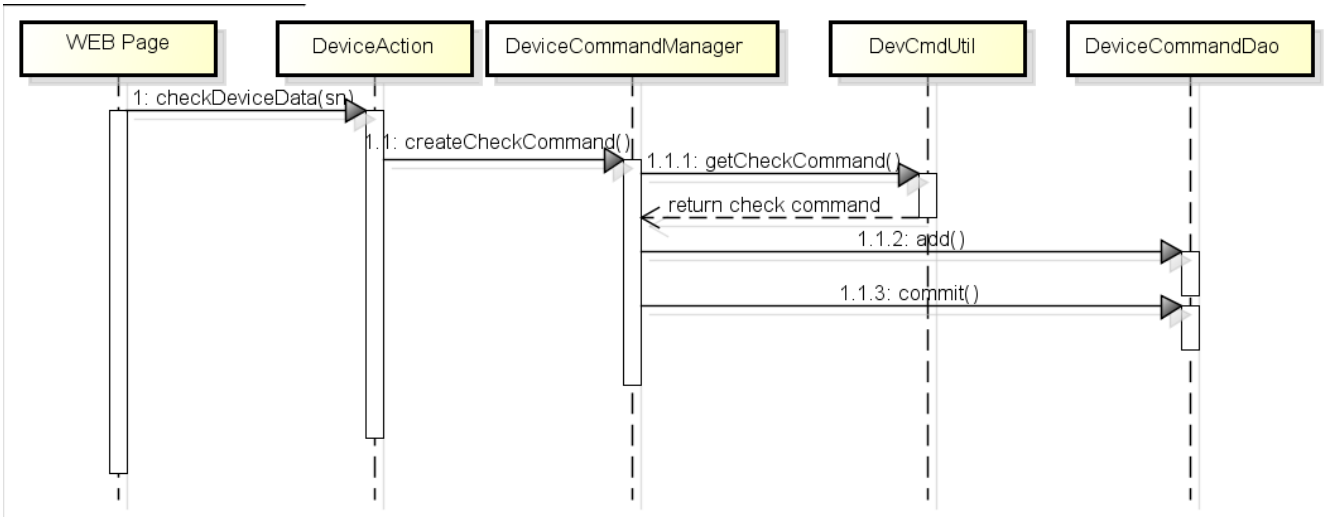
This command checks for data updates for devices. The operations involving this command in the Demo include:

- **Device management**

Re-obtaining device data (using the CHECK command with Stamp being set to 0).

Checking for and transferring data (using the CHECK command with Stamp not being set to 0).

The sequence of re-obtaining device data is taken as an example. See the following figure.



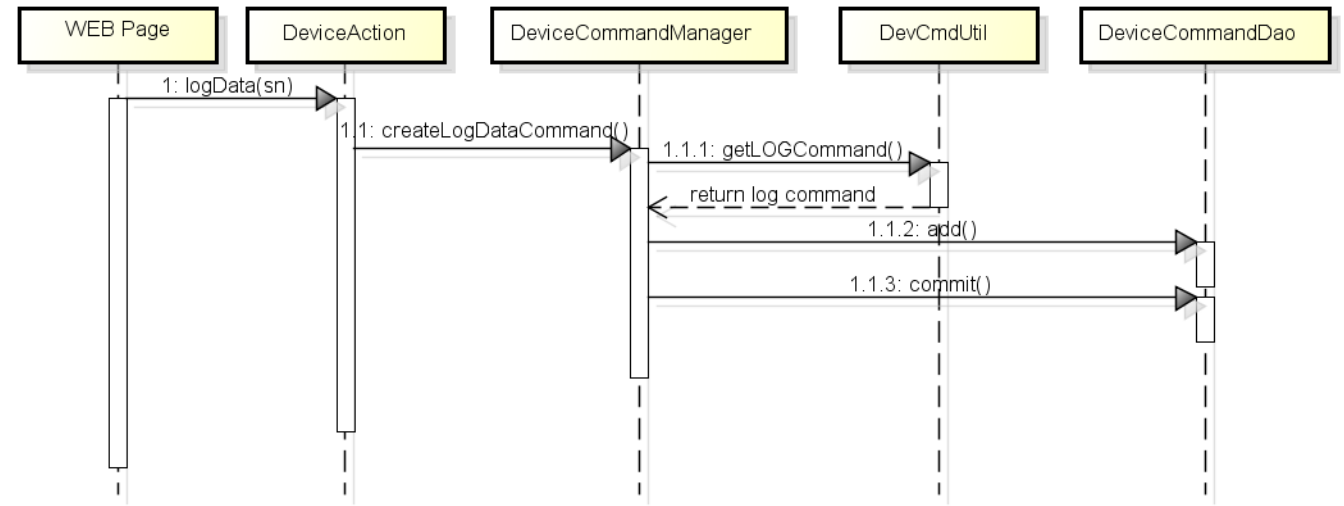
6.5.22 Check Command - Checking for and Transferring New Data

This command checks for and transfers new data from devices. The operations involving this command in the Demo include:

- **Device management**

Checking for and transferring new data (using the LOG command).

Sequence diagram:



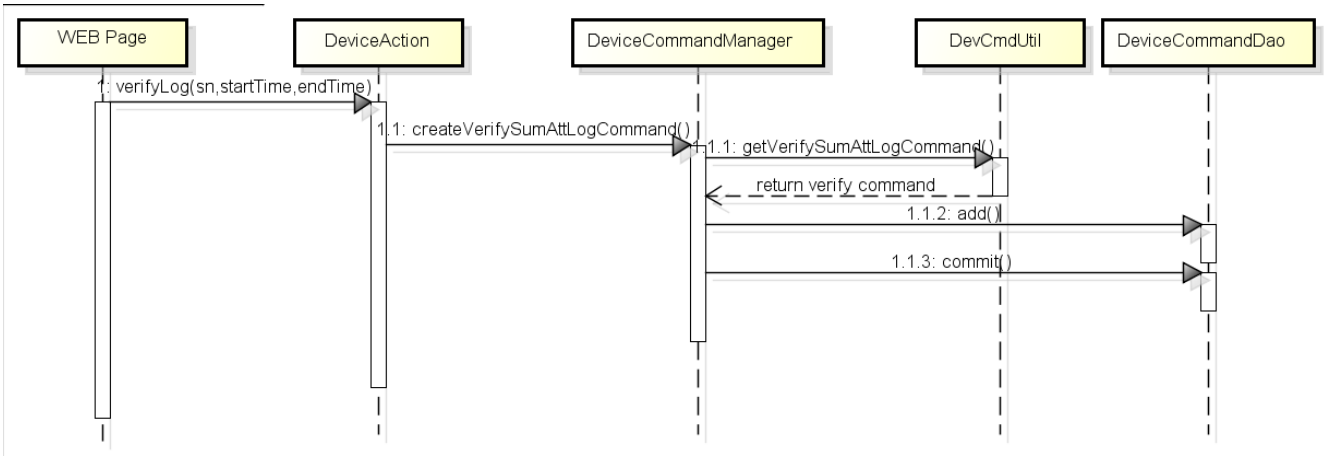
6.5.23 Check Command - Automatically Verifying Attendance Data

This command automatically verifies attendance data for devices. The operations involving this command in the Demo include:

- **Device management**

Automatically verifying attendance data (using the VERIFY SUM ATTLOG command).

Sequence diagram:



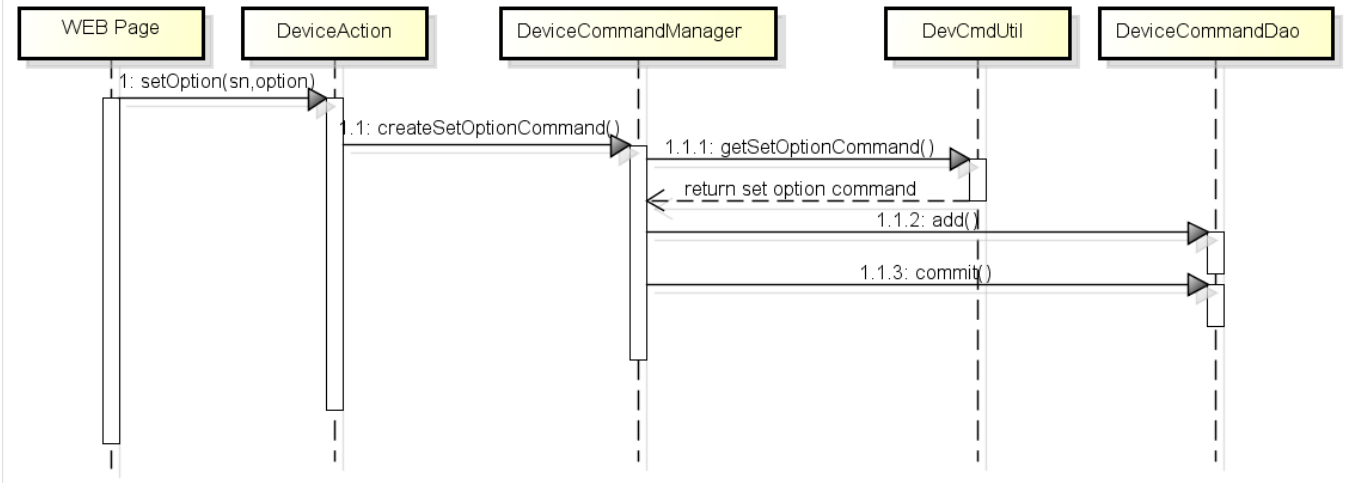
6.5.24 Option Setup Command - Setting Client Options

This command sets client options for devices. The operations involving this command in the Demo include:

- **Device management**

Setting client options (using the SET OPTION command).

Sequence diagram:



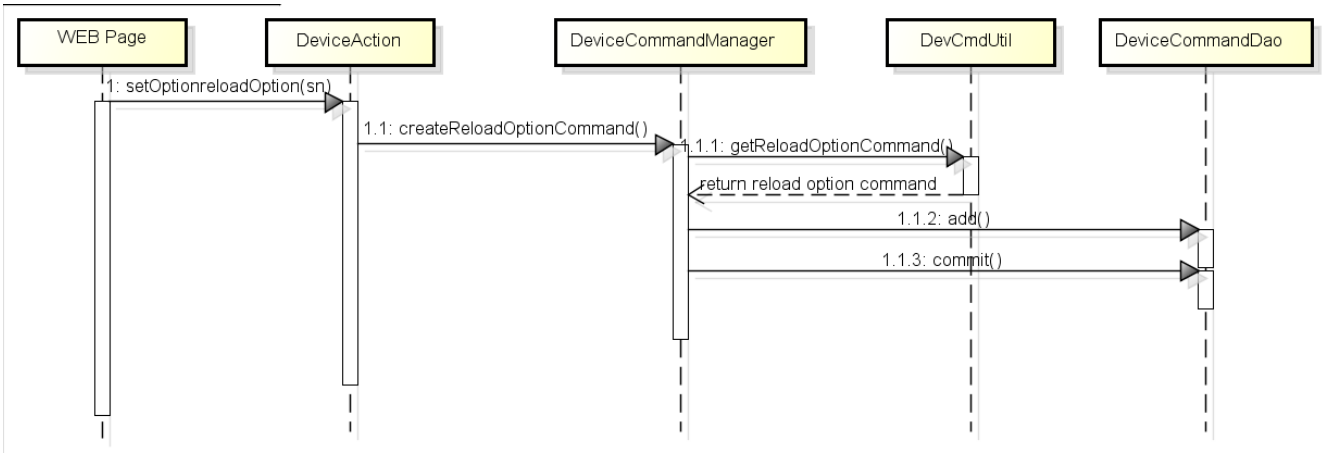
6.5.25 Option Setup Command - Reloading Client Options

This command reloads client options for devices. The operations involving this command in the Demo include:

- **Device management**

Reloading client options (using the RELOAD OPTIONS command).

Sequence diagram:



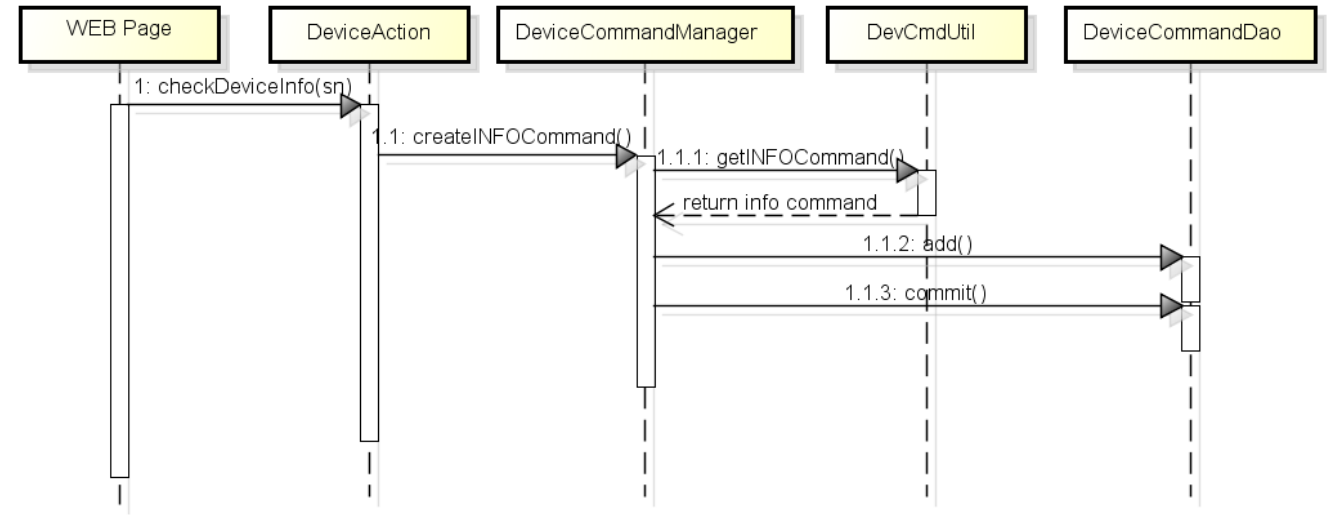
6.5.26 Option Setup Command - Sending Client Information to the Server

This command sends client information to the server for devices. The operations involving this command in the Demo include:

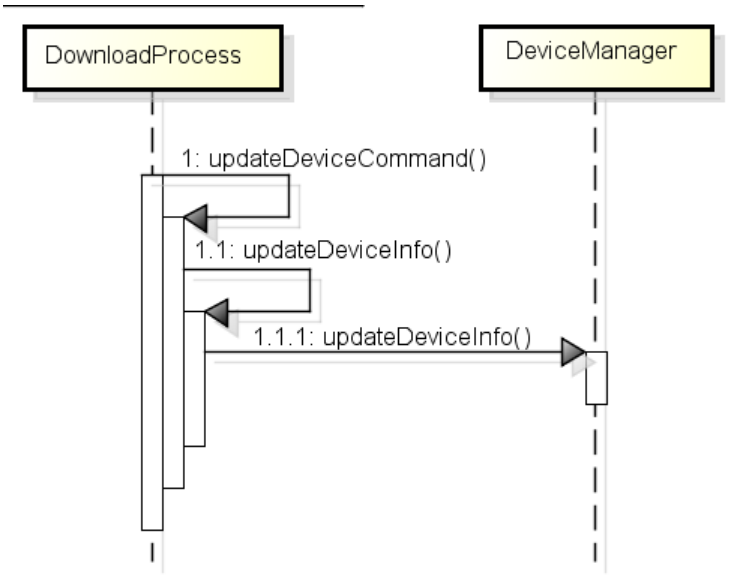
- **Device management**

Updating device information (using the INFO command).

Sequence diagram:



Returning a command execution result:



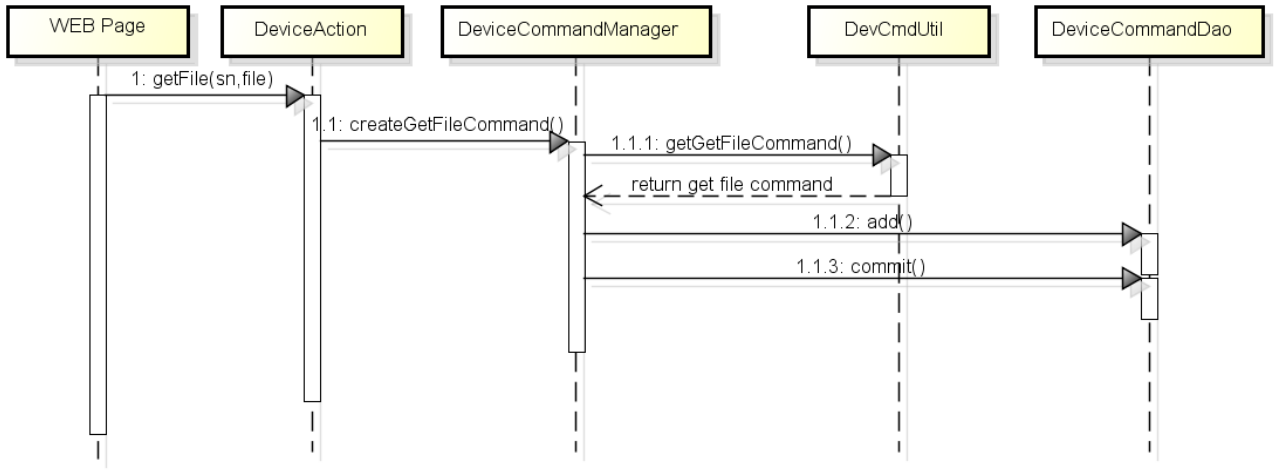
6.5.27 File Command - Obtaining Files from a Client

This command obtains files from a client for devices. The operations involving this command in the Demo include:

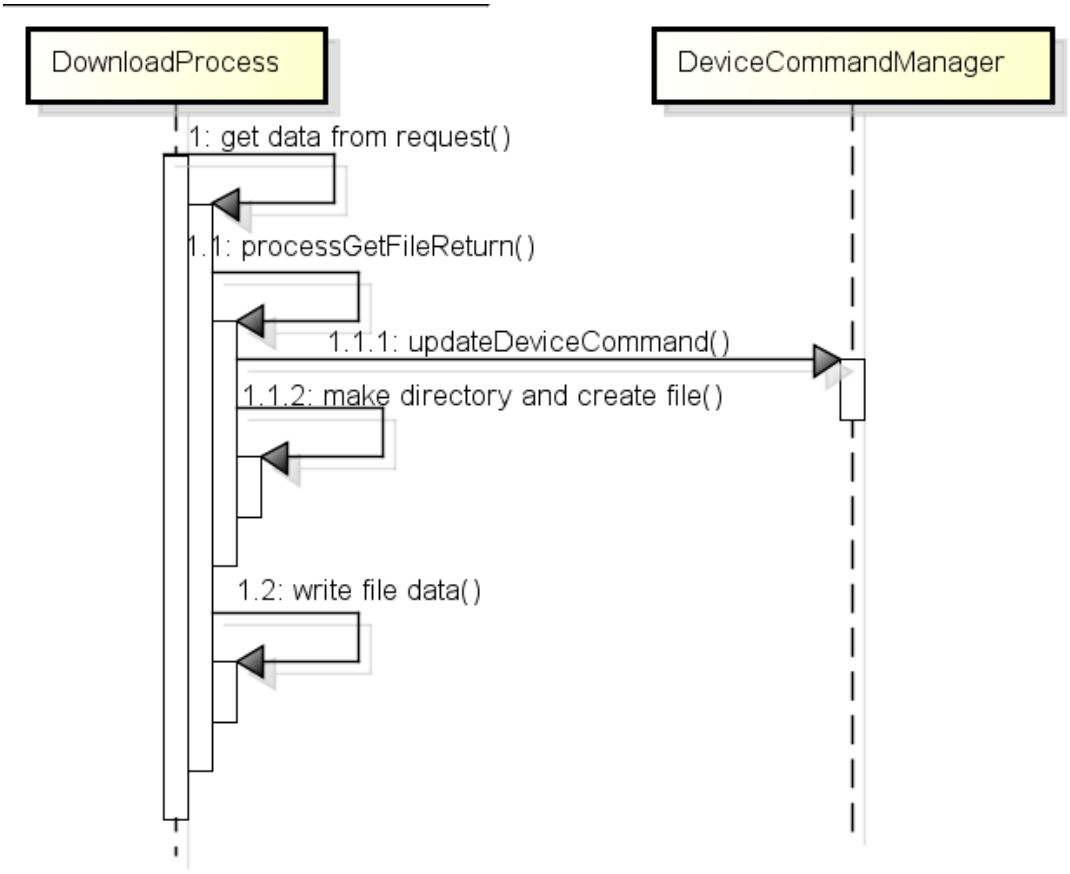
- **Device management**

Obtaining files from a client (using the GetFile command).

Sequence diagram:



Returning a command execution result:



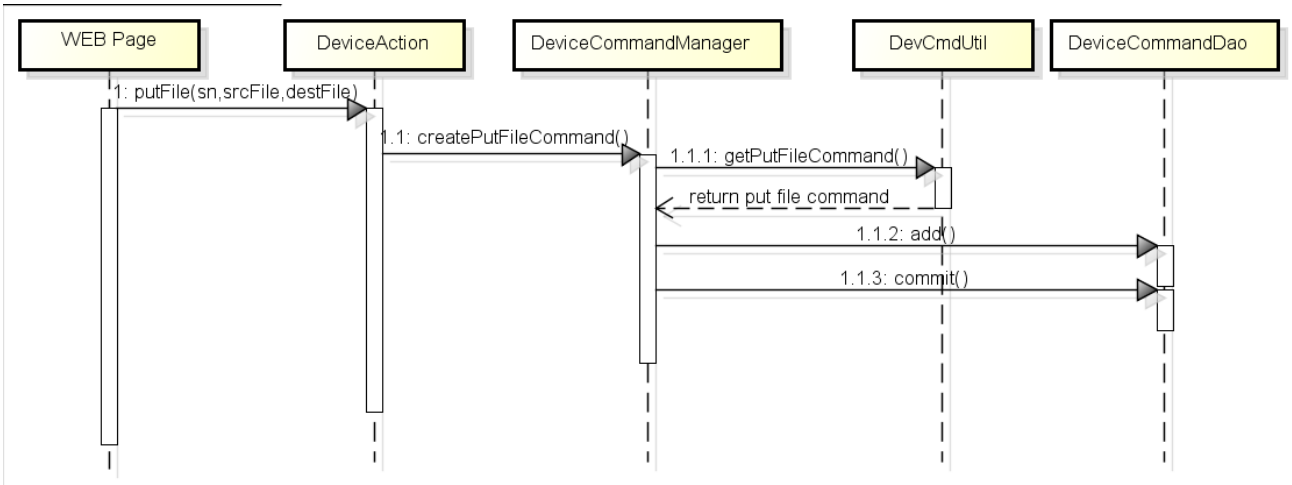
6.5.28 File Command - Sending Files to a Client

This command sends files to a client for devices. The operations involving this command in the Demo include:

- **Device management**

Sending files to a client (using the PutFile command).

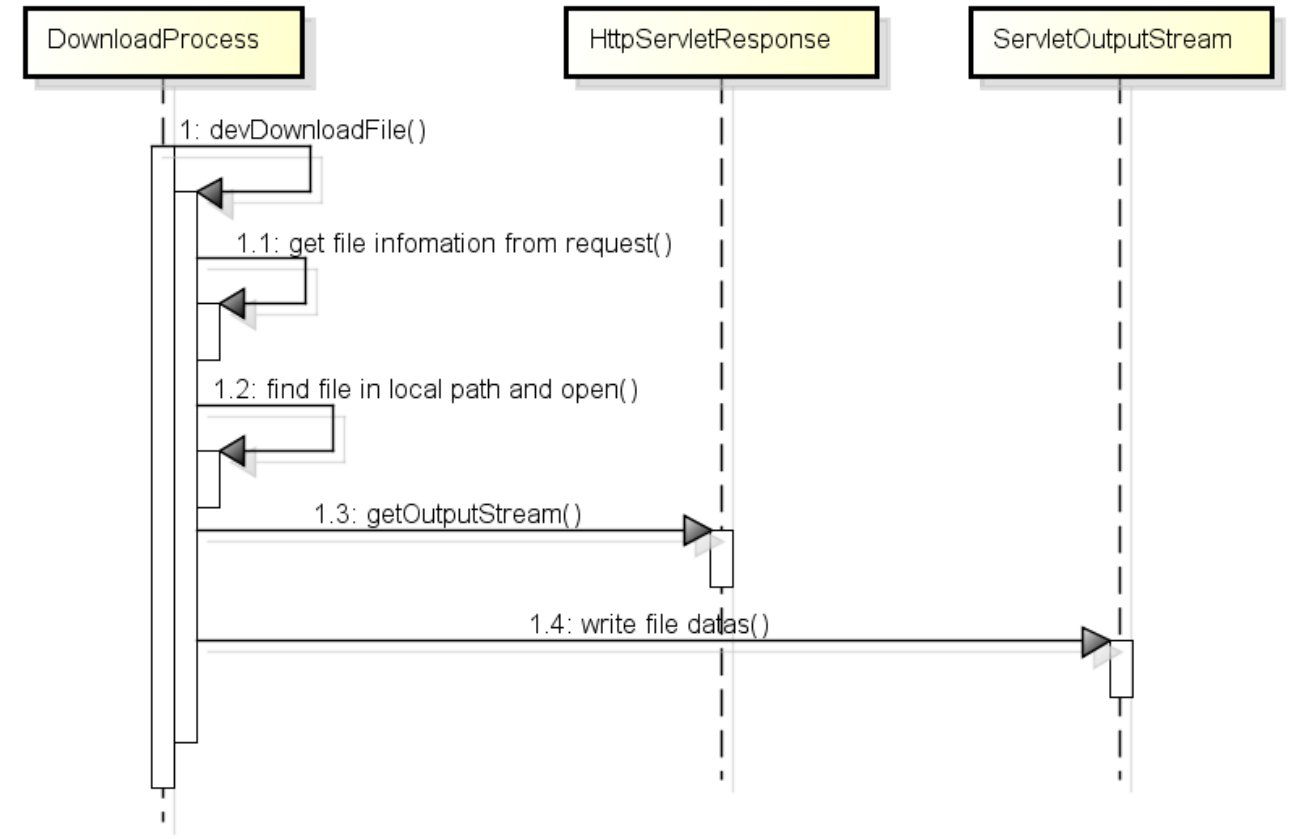
Sequence diagram:



Returning a command execution result:

- If files have absolute addresses, clients use the addresses to download the files directly.

➢ If files have relative addresses, the download process is as follows:



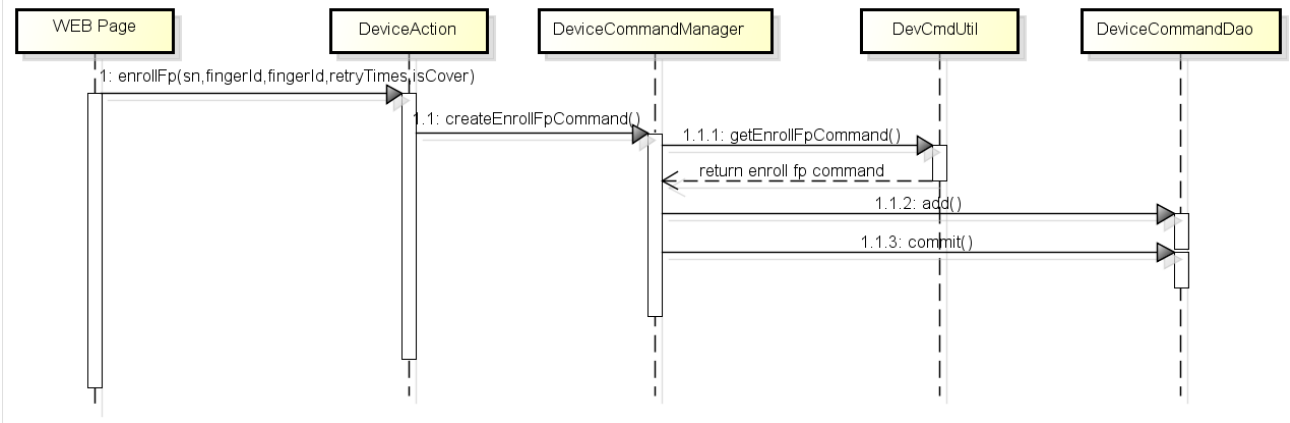
6.5.29 Remote Registration Command - Remotely Registering User Fingerprints

This command remotely registers user fingerprints for devices. The operations involving this command in the Demo include:

➢ **Device management**

Registering user fingerprints (using the ENROLL_FP command).

Sequence diagram:



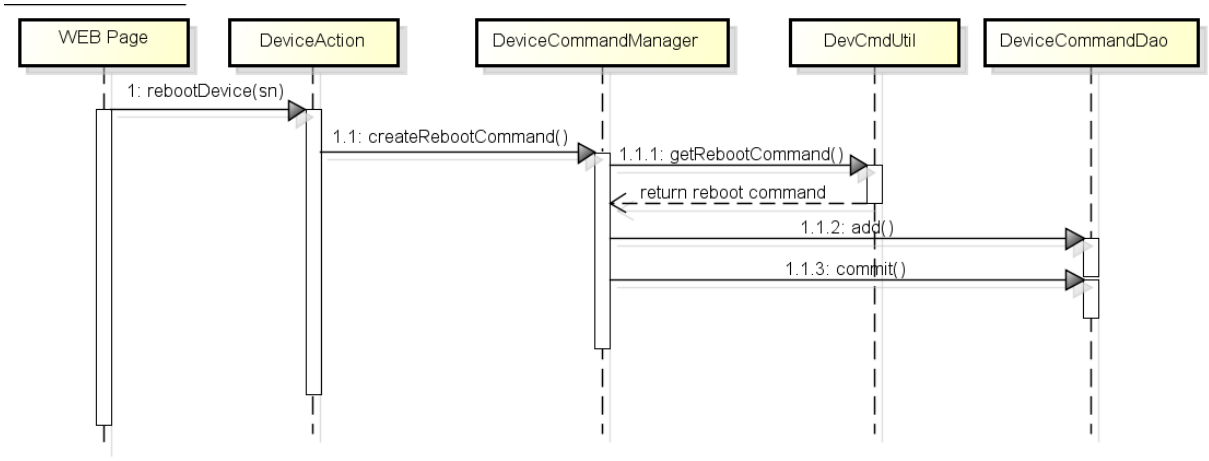
6.5.30 Control Command - Restarting a Client

This command restarts a client for devices. The operations involving this command in the Demo include:

- Device management

Restarting a client (using the REBOOT command).

Sequence diagram:



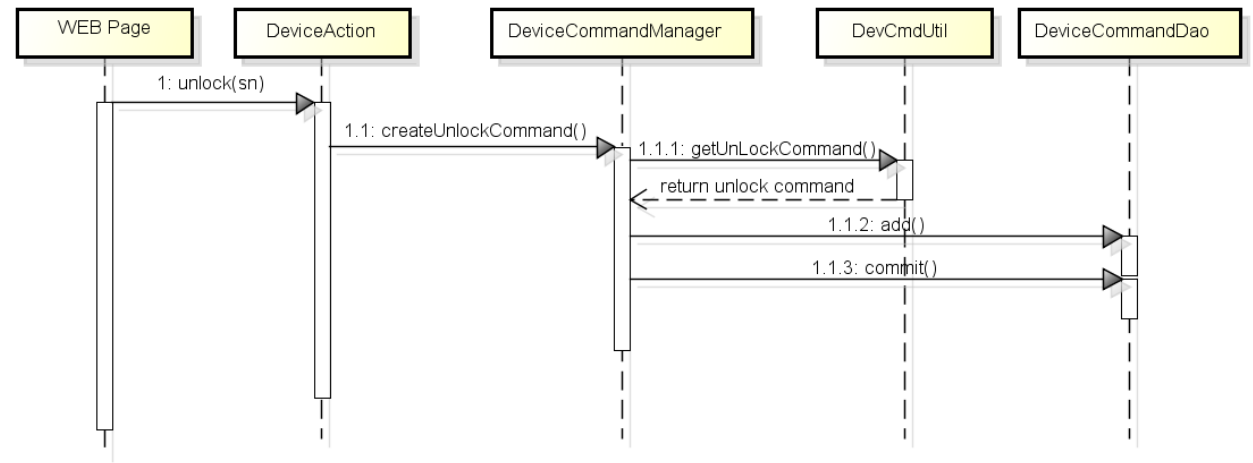
6.5.31 Control Command - Outputting the Unlocking Signal

This command outputs the unlocking signal for devices. The operations involving this command in the Demo include:

- Device management

Outputting the unlocking signal (using the AC_UNLOCK command).

Sequence diagram:



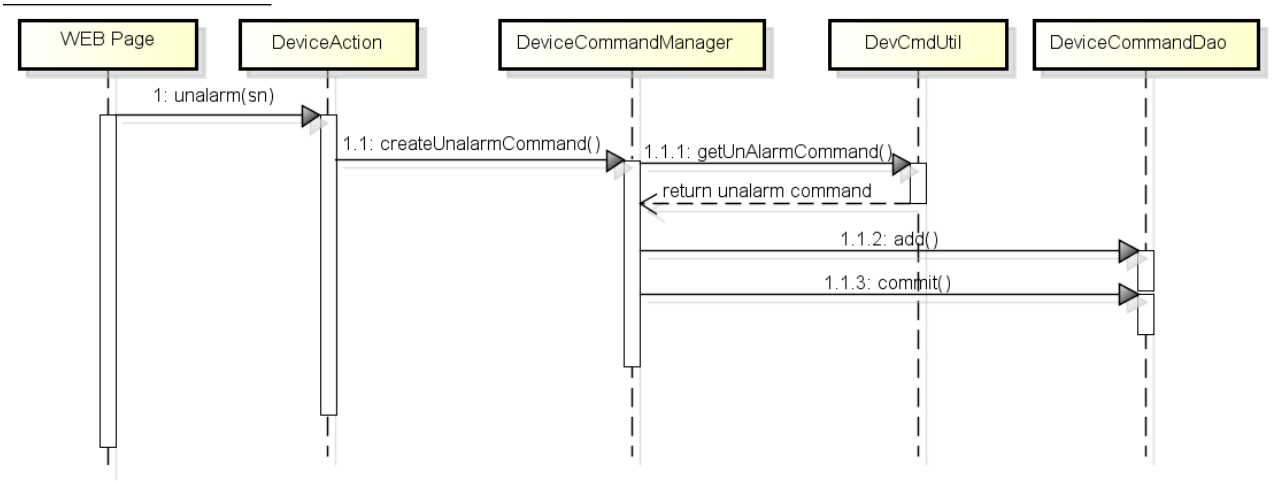
6.5.32 Control Command - Cancelling Alarm Signal Output

This command cancels alarm signal output for devices. The operations involving this command in the Demo include:

➢ Device management

Cancelling alarm signal output (using the AC_UNALARM command).

Sequence diagram:



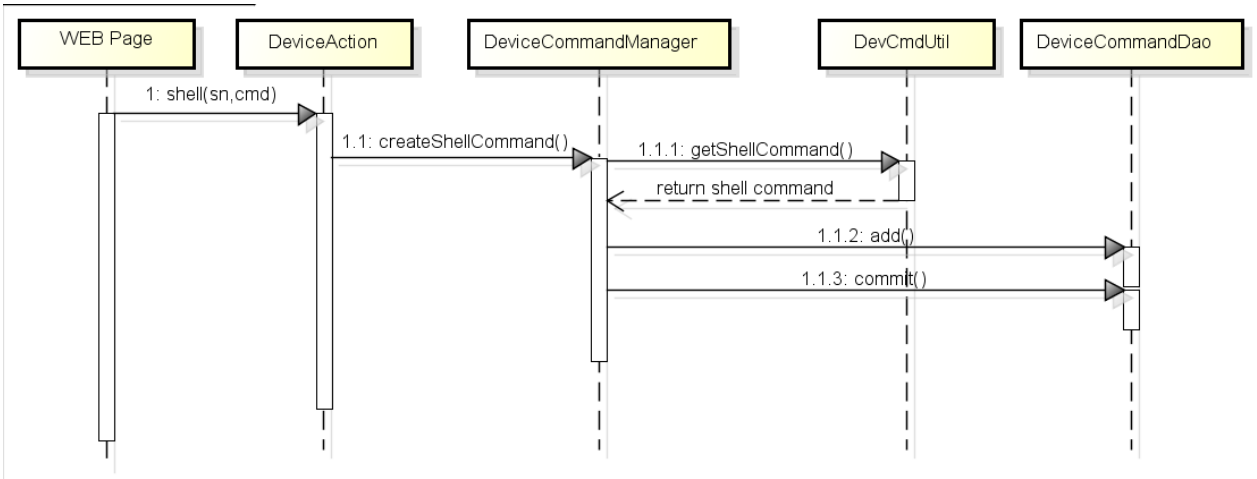
6.5.33 Other Command - Executing a System Command

This command executes a system command for devices. The operations involving this command in the Demo include:

➢ Device management

Executing a system command (using the SHELL command).

Sequence diagram:



6.6 Registering Attendance Remotely

Device request command

```
GET /iclock/cdata?SN=${SerialNumber}&table=RemoteAtt&PIN=${XXX}
```

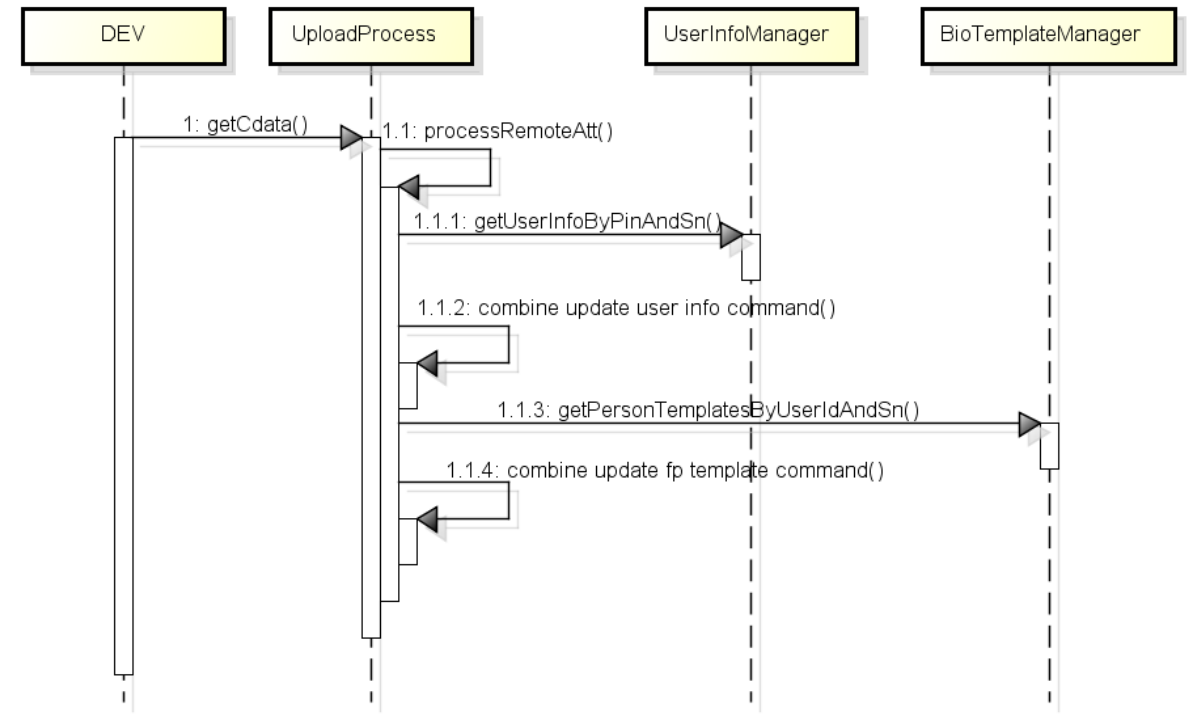
Key parameters:

SN: device SN.

table: operation table. Its value must be RemoteAtt.

PIN: user work ID.

Sequence diagram:



7. UI Function Design

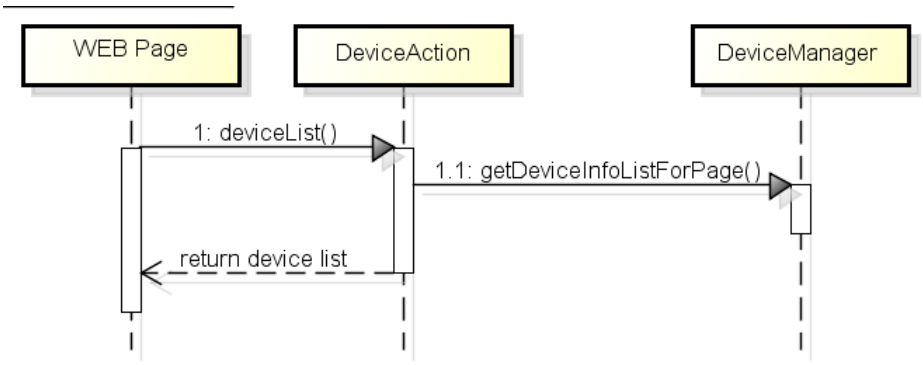
7.1 Managing Devices

This function displays and manages device information and delivers device commands to devices based on devices.

7.1.1 Device List

This function obtains a device list from the server database and displays the list.

Sequence diagram:



7.1.2 Editing Device Information

An operator can click a device SN in the device list to access the page for editing device information, and edit the following information:

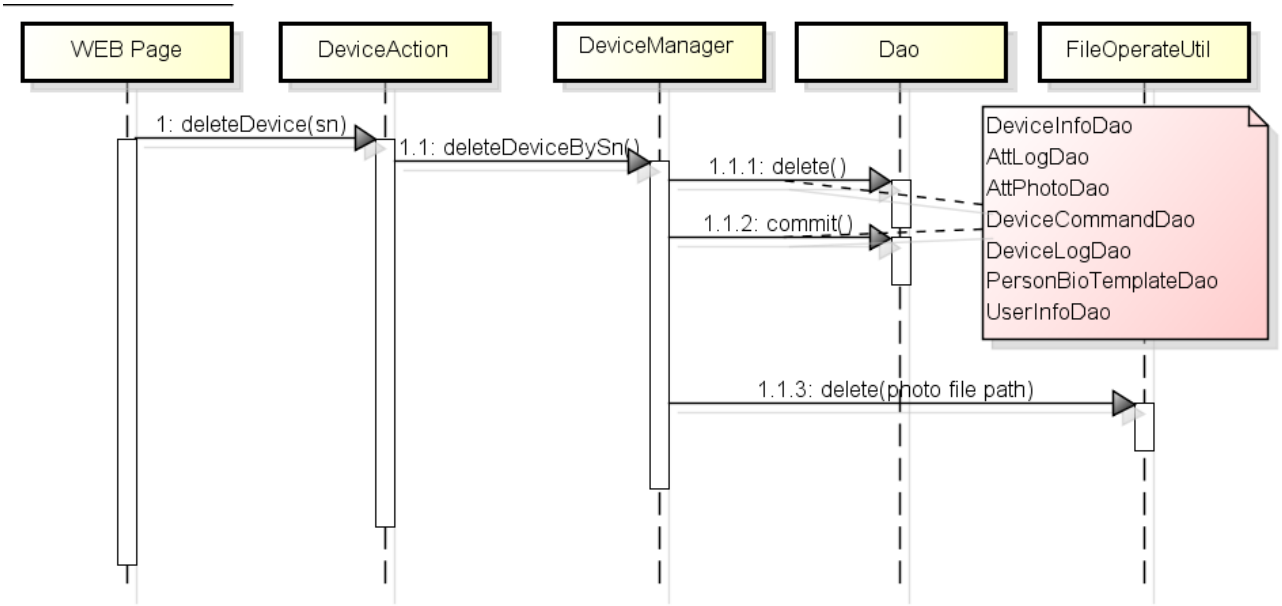
- Device name
- Transfer time
- Update interval

7.1.3 Deleting Devices

This function deletes device information from the server along with all the data related to the deleted device. The data includes:

- Device information
- User information
- User biometric identification template information
- Attendance logs
- Attendance photos and files
- Device commands
- Device operation logs

Sequence diagram:

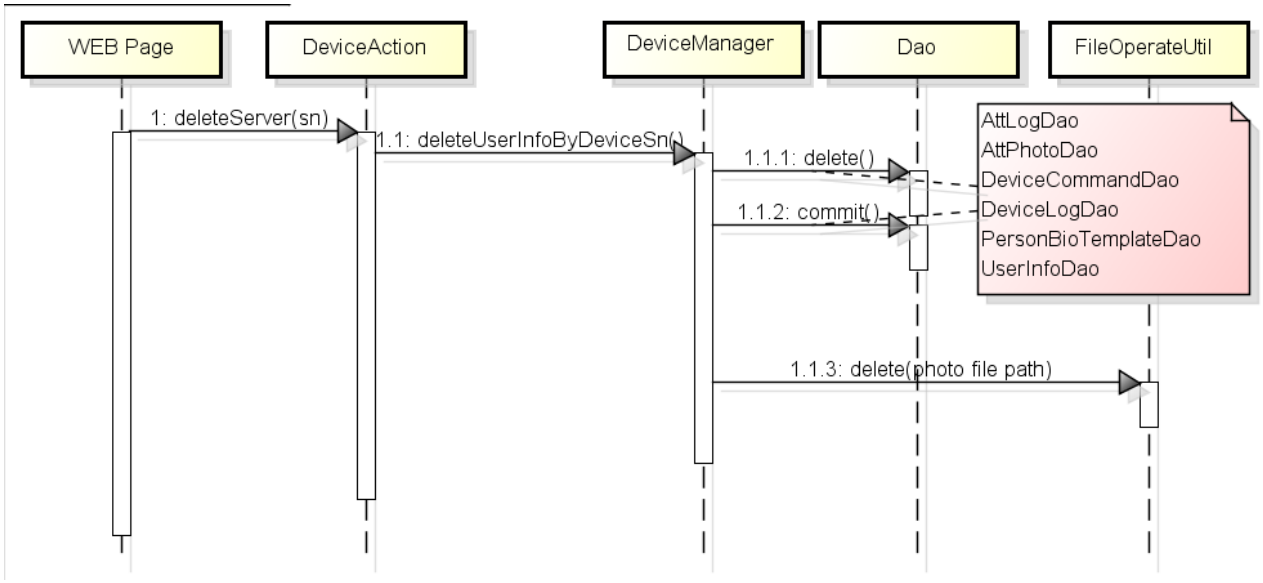


7.1.4 Deleting Data from the Server (Including User, Fingerprint, Face, and Attendance Data)

This function deletes device data from the server, including:

- User information
- User biometric identification template information
- Attendance logs
- Attendance photos and files
- Device commands
- Device operation logs

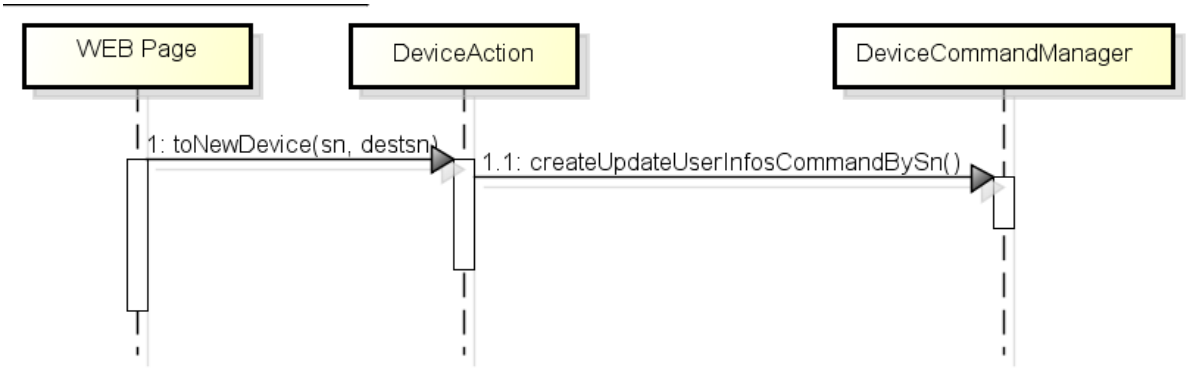
Sequence diagram:



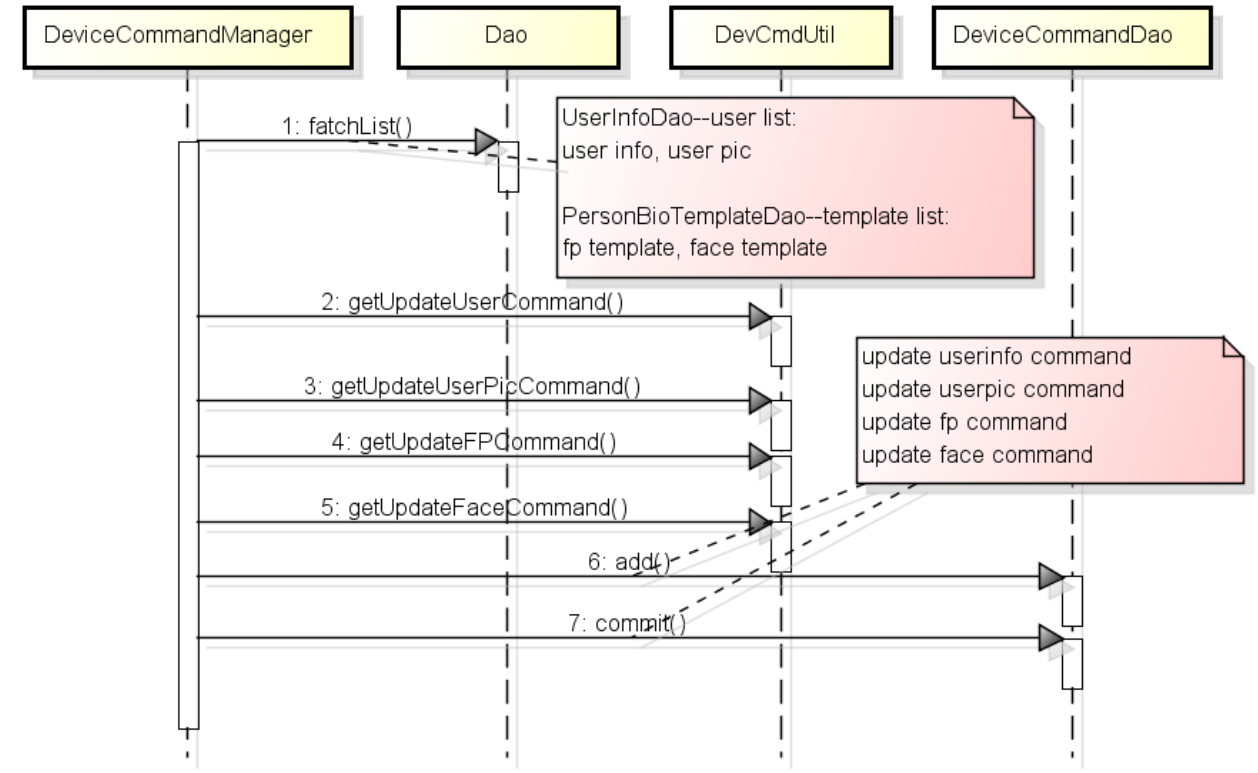
7.1.5 Backing Up Data to Other Devices (Using the DATA UPDATE Command to Back Up User, Fingerprint, Face, and User Photo Data)

This function transfers all user data from a specified device to a new device and generates a device command.

Sequence diagram:



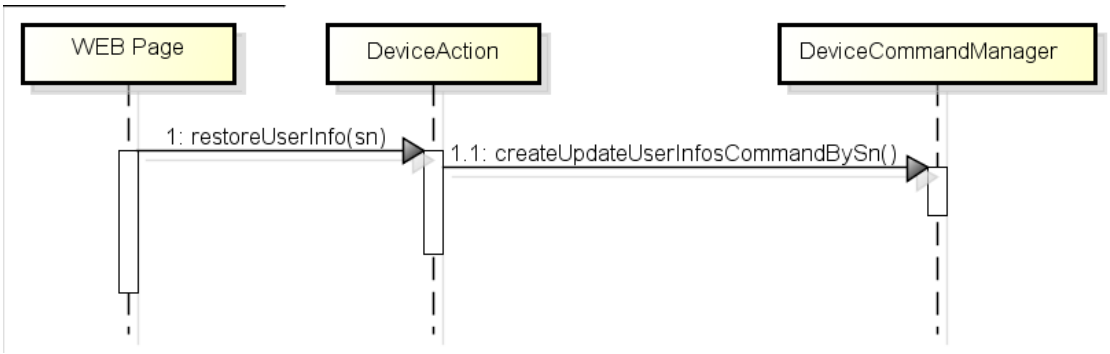
Sequence diagram of the DeviceCommandManager$createUpdateUserInfosCommandBySn method:



7.1.6 Restoring Data (Using the DATA UPDATE Command to Restore User, Fingerprint, Face, and User Photo Data)

This function delivers all user data of a specified device from the server to the device and generates a device command.

Sequence diagram:



For the sequence diagram of the DeviceCommandManager$createUpdateUserInfosCommandBySn method, see Section 7.1.6 "Restoring Data (Using the DATA UPDATE Command to Restore User, Fingerprint, Face, and User Photo Data)."

7.1.7 Querying Attendance Logs (Using the DATA QUERY ATTLOG Command)

See Section 6.5.14 "DATA QUERY Command - Attendance Logs."

7.1.8 Querying Attendance Photos (Using the DATA QUERY ATTPHOTO Command)

See Section 6.5.15 "DATA QUERY Command - Attendance Photos."

7.1.9 Querying User Information (Using the DATA QUERY USERINFO Command)

See Section 6.5.16 "DATA QUERY Command - User Information."

7.1.10 Querying the Fingerprint Data Template (Using the DATA QUERY FINGERTMP Command)

See Section 6.5.17 "DATA QUERY Command - Fingerprint Data Template."

7.1.11 Clearing All Data (Using the CLEAR DATA Command)

See Section 6.5.20 "CLEAR Command - All Data."

7.1.12 Clearing Attendance Logs (Using the CLEAR LOG Command)

See Section 6.5.18 "CLEAR Command - Attendance Logs."

7.1.13 Clearing Attendance Photos (Using the CLEAR PHOTO Command)

See Section 6.5.19 "CLEAR Command - Attendance Photos."

7.1.14 Re-obtaining Device Data (Using the CHECK Command with Stamp Being Set to 0)

See Section 6.5.21 "Check Command - Checking for Data Updates."

7.1.15 Checking for and Transferring Data (Using the CHECK Command with Stamp Not Being Set to 0)

See Section 6.5.21 "Check Command - Checking for Data Updates."

7.1.16 Checking for and Transferring New Data (Using the LOG Command)

See Section 6.5.22 "Check Command - Checking for and Transferring New Data" section.

7.1.17 Automatically Verifying Attendance Data (Using the VERIFY SUM ATTLOG Command)

See Section 6.5.23 "Check Command - Automatically Verifying Attendance Data" section.

7.1.18 Obtaining Files from a Client (Using the GetFile Command)

See Section 6.5.27 "File Command - Obtaining Files from a Client."

7.1.19 Sending Files to a Client (Using the PutFile Command)

See Section 6.5.28 "File Command - Sending Files to a Client."

7.1.20 Updating Device Information (Using the INFO Command)

See Section 6.5.26 "Option Setup Command - Sending Client Information to the Server."

7.1.21 Setting Client Options (Using the SET OPTION Command)

See Section 6.5.24 "Option Setup Command - Setting Client Options."

7.1.22 Reloading Client Options (Using the RELOAD OPTIONS Command)

See Section 6.5.25 "Option Setup Command - Reloading Client Options."

7.1.23 Restarting a Client (Using the REBOOT Command)

See Section 6.5.30 "Control Command - Restarting a Client."

7.1.24 Outputting the Unlocking Signal (Using the AC_UNLOCK Command)

See Section 6.5.31 "Control Command - Outputting the Unlocking Signal."

7.1.25 Cancelling Alarm Signal Output (Using the AC_UNALARM Command)

See Section 6.5.32 "Control Command - Cancelling Alarm Signal Output."

7.1.26 Registering User Fingerprints (Using the ENROLL_FP Command)

See Section 6.5.29 "Remote Registration Command - Remotely Registering User Fingerprints" section.

7.1.27 Executing a System Command (Using the SHELL Command)

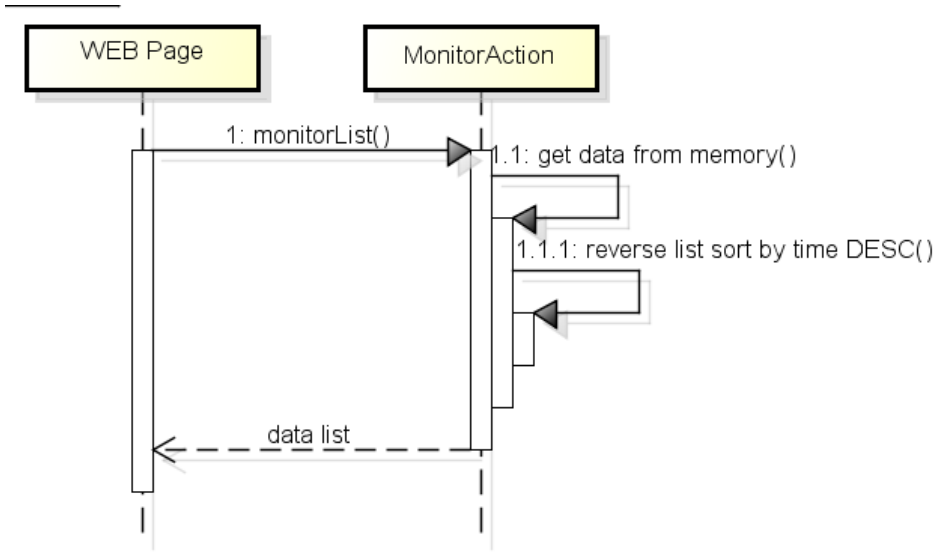
See Section 6.5.33 "Other Command - Executing a System Command."

7.2 Real-Time Monitoring

Device operation logs and attendance data are parsed and displayed in real time. The data is stored in the memory and lost when the server restarts.

The page is refreshed every five seconds.

Sequence diagram:



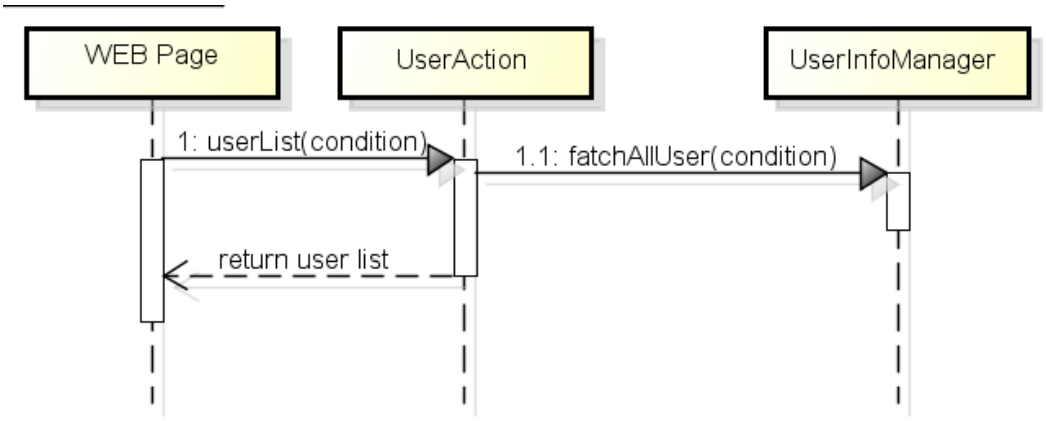
7.3 Managing Personnel

This function displays and manages personnel information and delivers device commands to devices based on personnel.

7.3.1 Querying Personnel

This function queries personnel data by the specified criteria and displays the data in a list. By default, data are queried without criteria.

Sequence diagram:

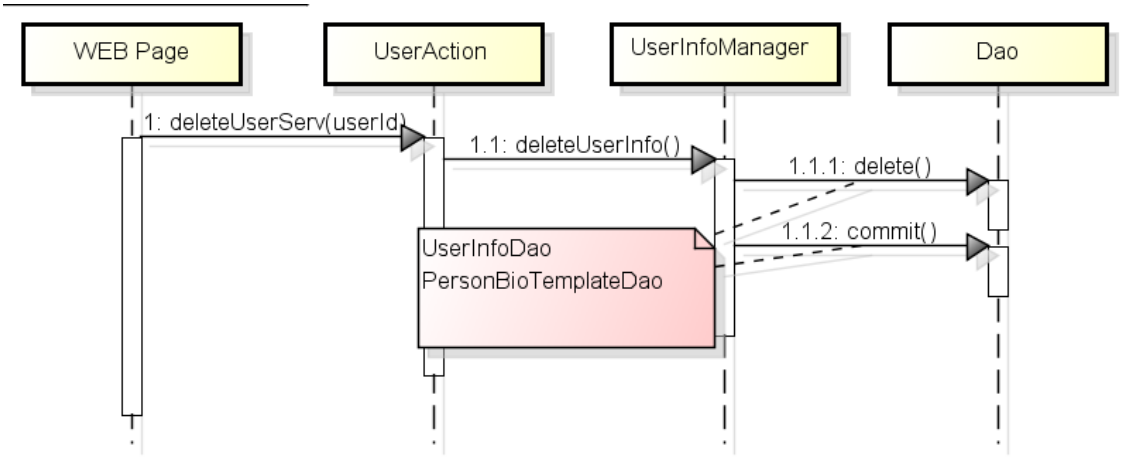


7.3.2 Deleting Personnel Information from the Server

This function deletes information about specified personnel from the server, including:

- Basic personnel information.
- User biometric identification template.

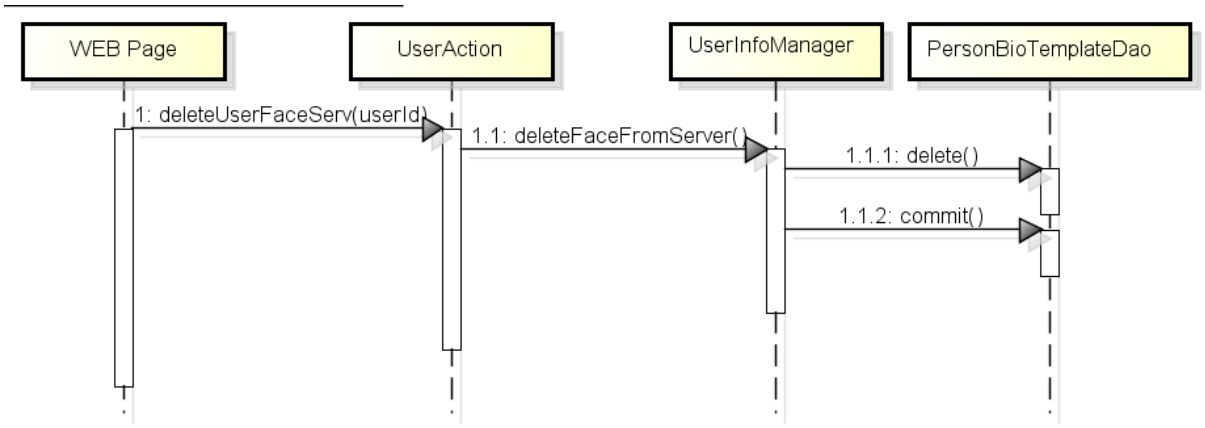
Sequence diagram:



7.3.3 Deleting the Face Data Template from the Server

This function deletes the face data template of specified personnel from the server.

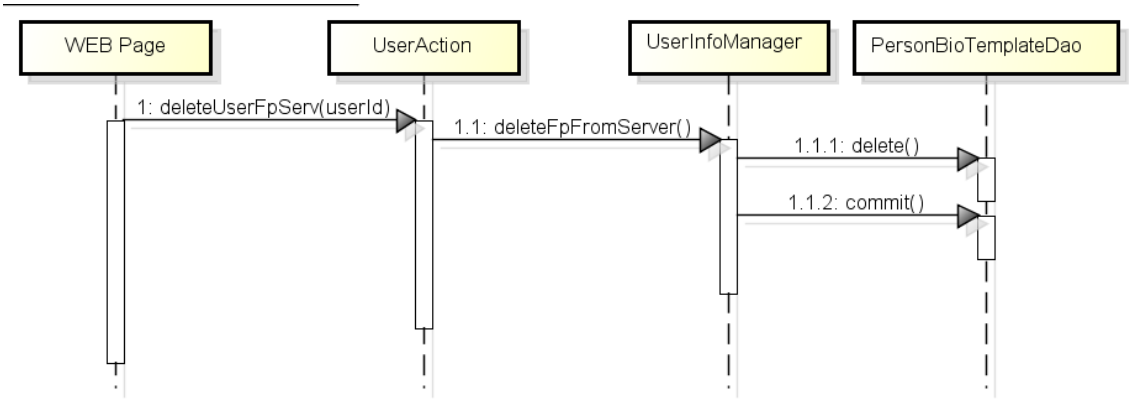
Sequence diagram:



7.3.4 Deleting the Fingerprint Data Template from the Server

This function deletes the fingerprint data template of specified personnel from the server.

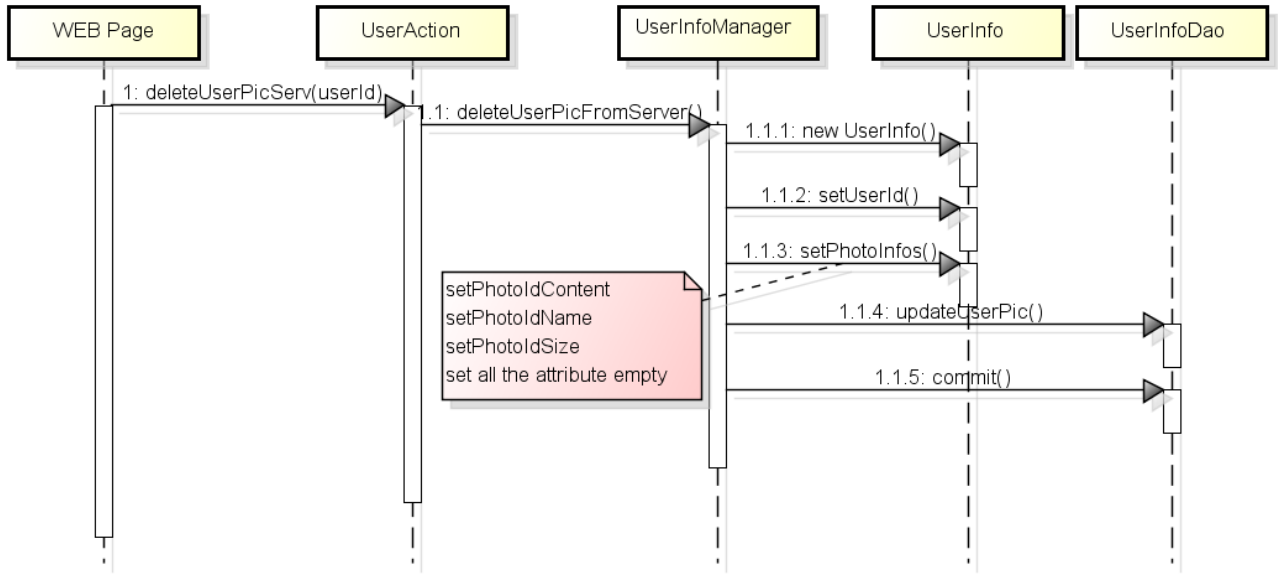
Sequence diagram:



7.3.5 Deleting User Photos from the Server

This function deletes user photos of specified personnel from the server. That is, it clears the user photo field information in user information.

Sequence diagram:



7.3.6 Sending Personnel Data to a Device (Using the DATA UPDATE Command to Send User, Fingerprint, Face, and User Photo Data)

This function sends data of specified personnel to the devices to which the personnel belong, including:

- Basic user information.
- Fingerprint data template.
- Face data template.
- User photos.

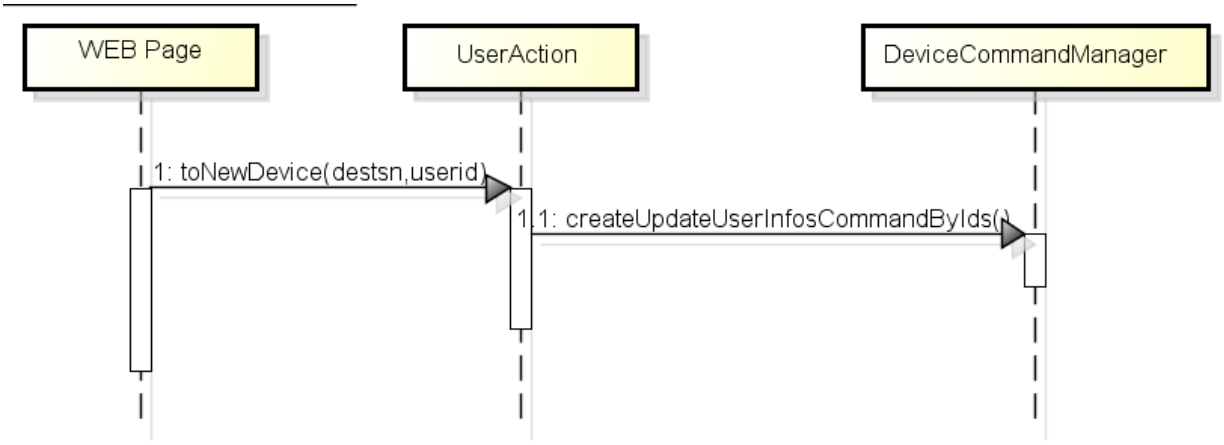
For the sequence diagram, see Section 6.5.3 "DATA UPDATE Command - User Information."

7.3.7 Migrating Personnel Data to a New Device (Using the DATA UPDATE Command to Migrate User, Fingerprint, Face, and User Photo Data)

This function sends data of specified personnel to a new device, including:

- Basic user information.
- Fingerprint data template.
- Face data template.
- User photos.

Sequence diagram:



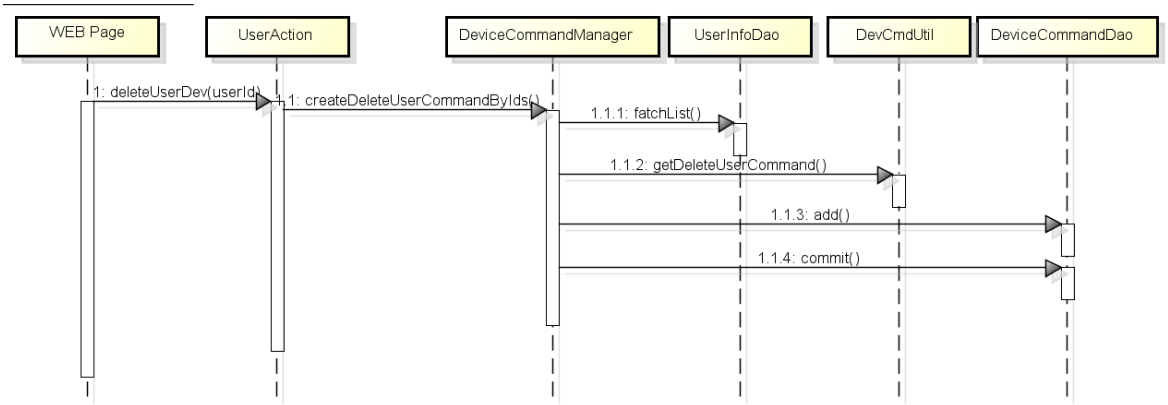
For the sequence diagram of the DeviceCommandManager$createUpdateUserInfosCommandByIds method, see Section 6.5.3 "DATA UPDATE Command - User Information."

7.3.8 Deleting Personnel Data from a Device (Using the DATA DELETE Command to Delete User, Fingerprint, Face, and User Photo Data)

This function deletes data of specified personnel from a device. When receiving the command, a device uses the function that it supports to delete the data of the specified personnel, including:

- Basic user information
- Fingerprint data template
- Face data template
- User photos

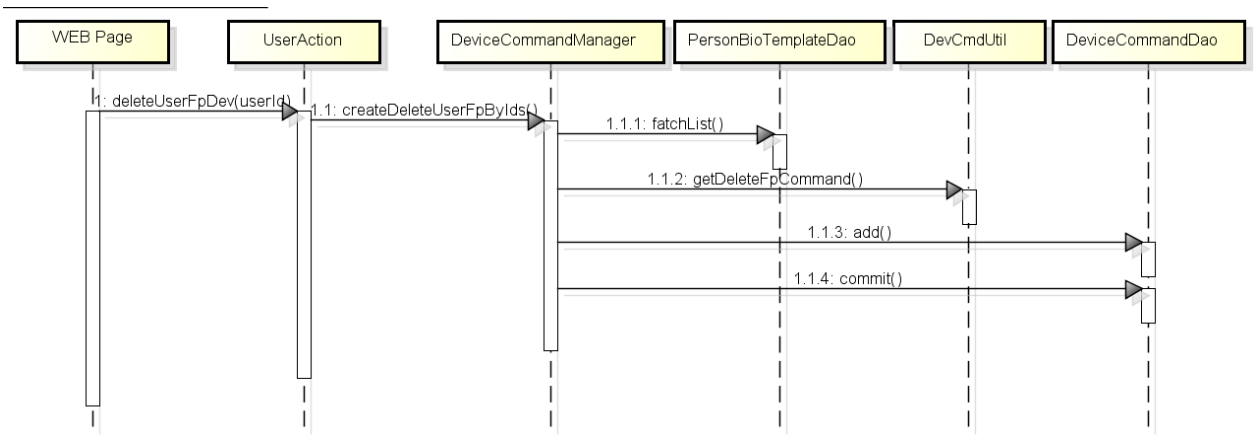
Sequence diagram:



You can also see Section 6.5.9 "DATA DELETE Command - User Information."

7.3.9 Deleting the Fingerprint Data Template from a Device (Using the DATA DELETE FINGERTMP Command)

See Section 6.5.10 "DATA DELETE Command - Fingerprint Data Template."



7.3.10 Deleting User Photos from a Device (Using the DATA DELETE USERPIC Command)

See Section 6.5.12 "DATA DELETE Command - User Photos."

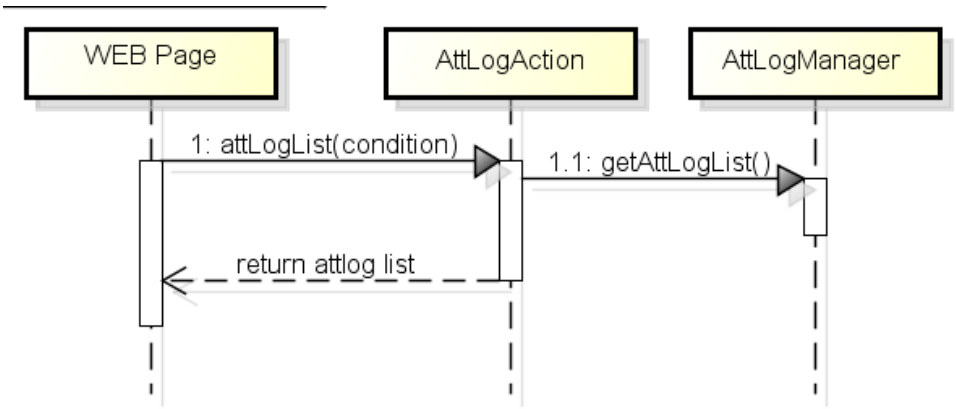
7.3.11 Deleting the Face Data Template from a Device (Using the DATA DELETE FACE Command)

See Section 6.5.11 "DATA DELETE Command - Face Data Template."

7.4 Attendance Logs

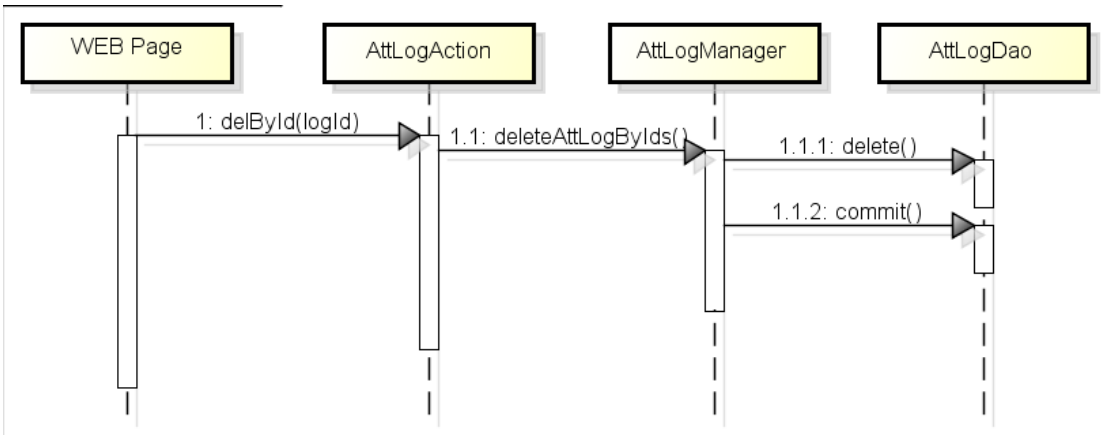
7.4.1 Querying Attendance Logs

This function queries attendance logs by the specified criteria and displays the logs in a list. By default, logs are queried without criteria.



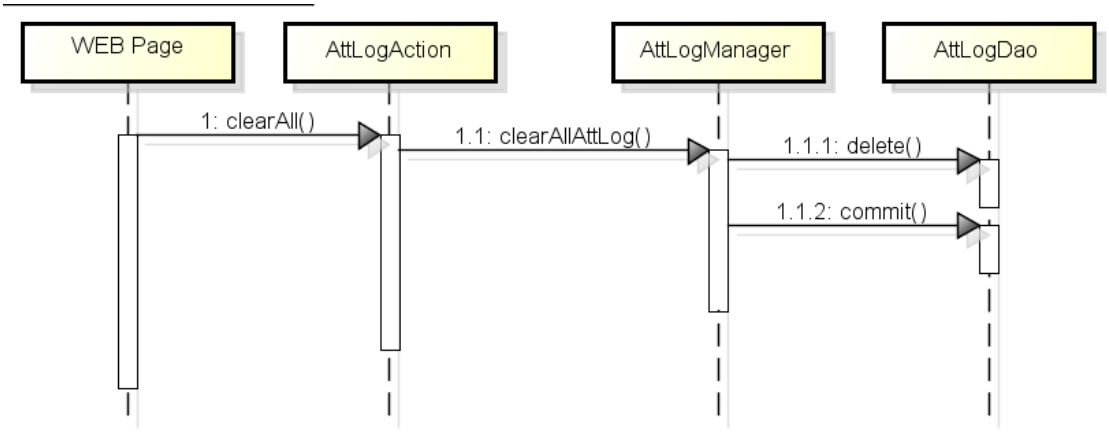
7.4.2 Deleting Selected Data

This function deletes the attendance logs selected in the list.



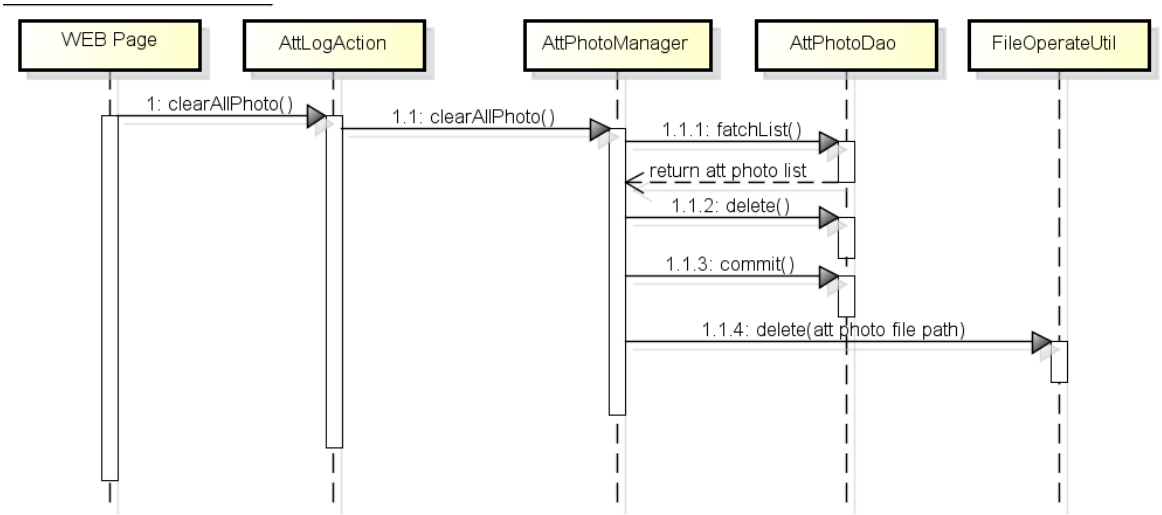
7.4.3 Clearing All Attendance Logs

This function clearing all attendance logs from the server database.



7.4.4 Clearing All Attendance Photos

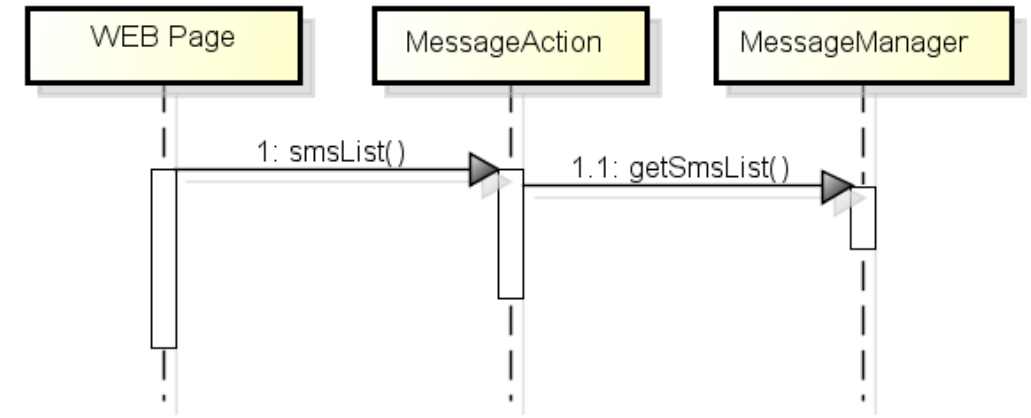
This function clears all attendance photos and image files from the server database.



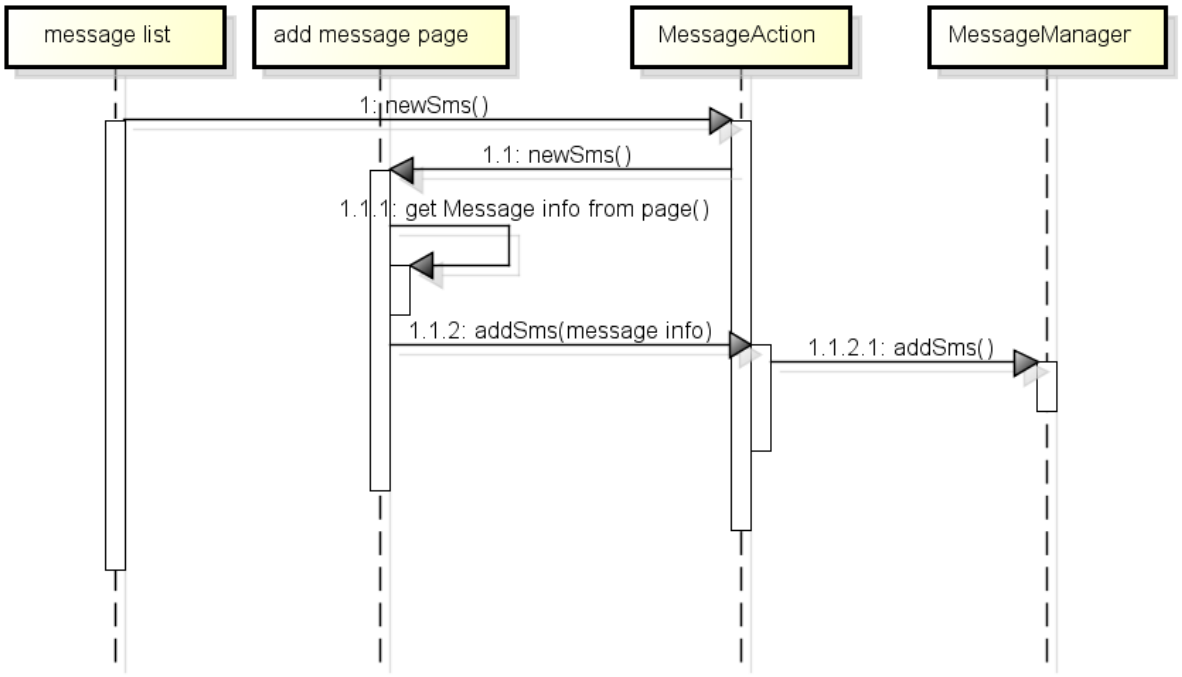
7.5 SMS Messages

7.5.1 SMS Message List

This function obtains SMS messages from the server and displays them in a list.

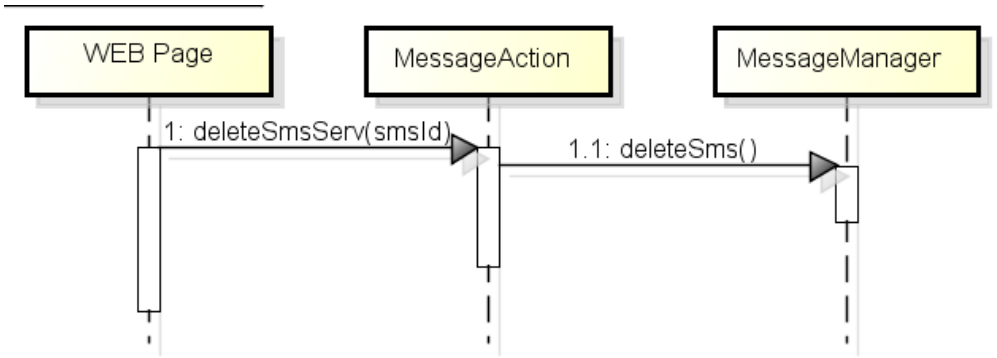


7.5.2 Adding an SMS Message



7.5.3 Deleting Selected SMS Messages

This function deletes the SMS messages selected in the SMS message list from the server.



7.5.4 Sending SMS Messages to Devices (Using the DATA UPDATE SMS Command)

See Section 6.5.7 "DATA UPDATE Command - SMS Messages."

7.5.5 Sending Personal SMS Messages to Devices (Using the DATA UPDATE USER_SMS Command)

See Section 6.5.8 "DATA UPDATE Command - Personal SMS Messages."

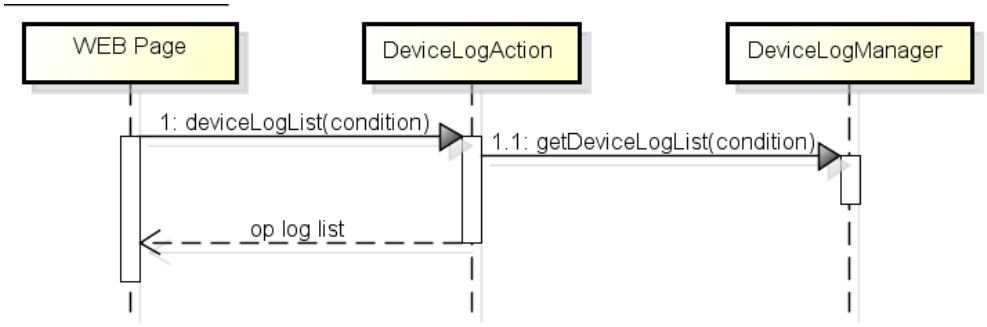
7.5.6 Deleting SMS Messages from a Device (Using the DATA DELETE SMS Command)

See Section 6.5.13 "DATA DELETE Command - SMS Messages" section.

7.6 Device Operation Logs

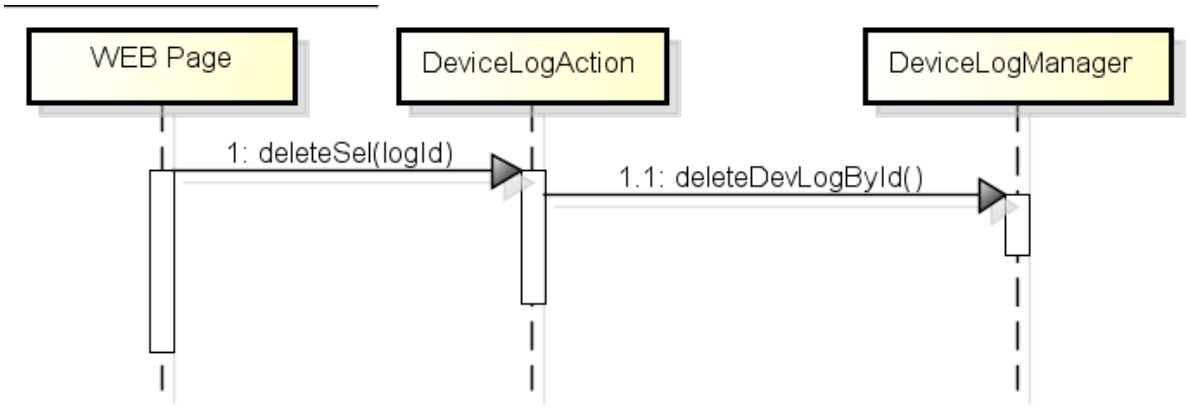
7.6.1 Querying Device Operation Logs

This function queries device operation logs by the specified criteria and displays the logs in a list. By default, logs are queried without criteria.



7.6.2 Deleting Selected Device Operation Logs

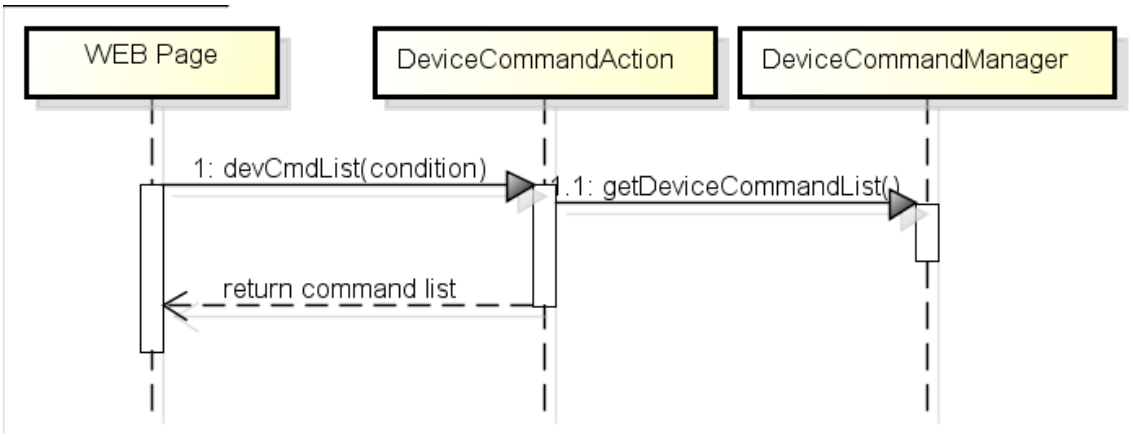
This function deletes the selected device operation logs selected in the list from the server.



7.7 Device Commands

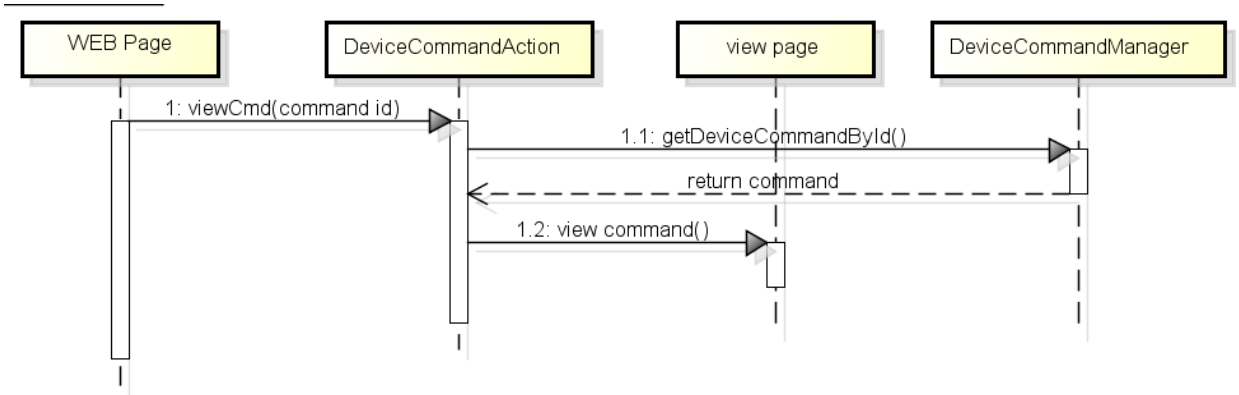
7.7.1 Querying Device Commands

This function queries device commands by the specified criteria and displays the commands in a list. By default, device commands are queried without criteria.



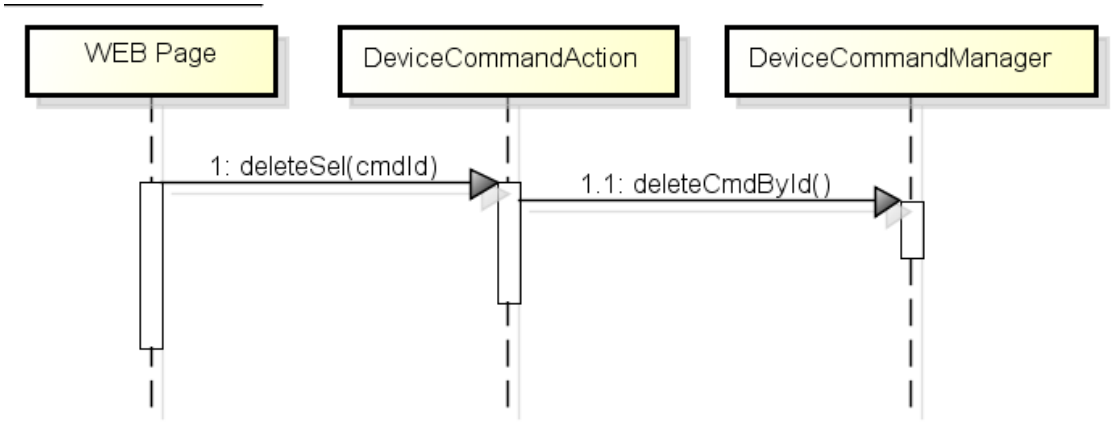
7.7.2 Viewing Command Details

An operator can click a command ID in a device command list to access the command details page.



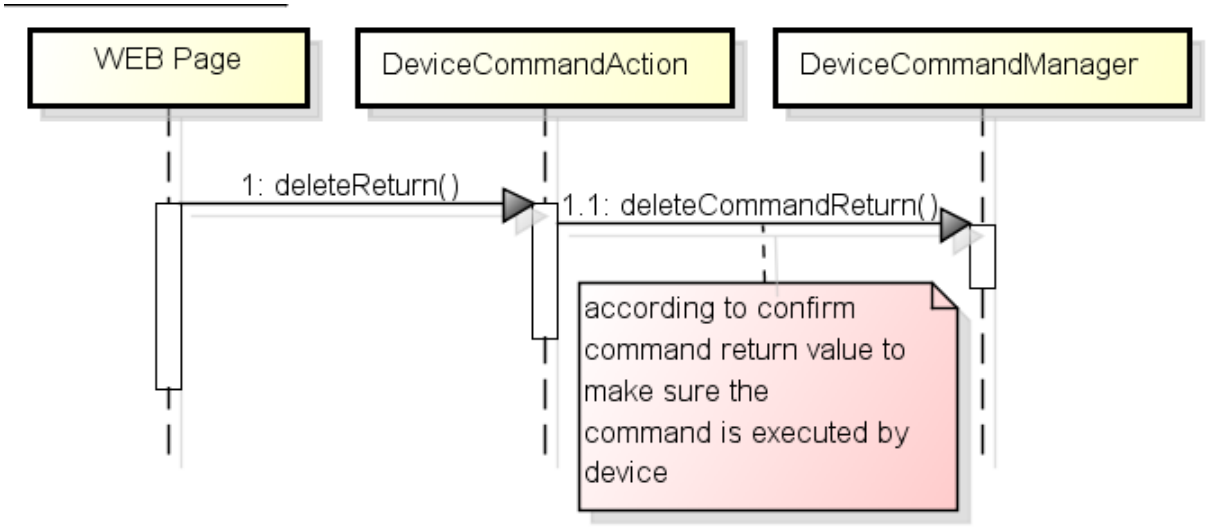
7.7.3 Deleting Selected Data

This function deletes the device commands selected in a device command list from the server.



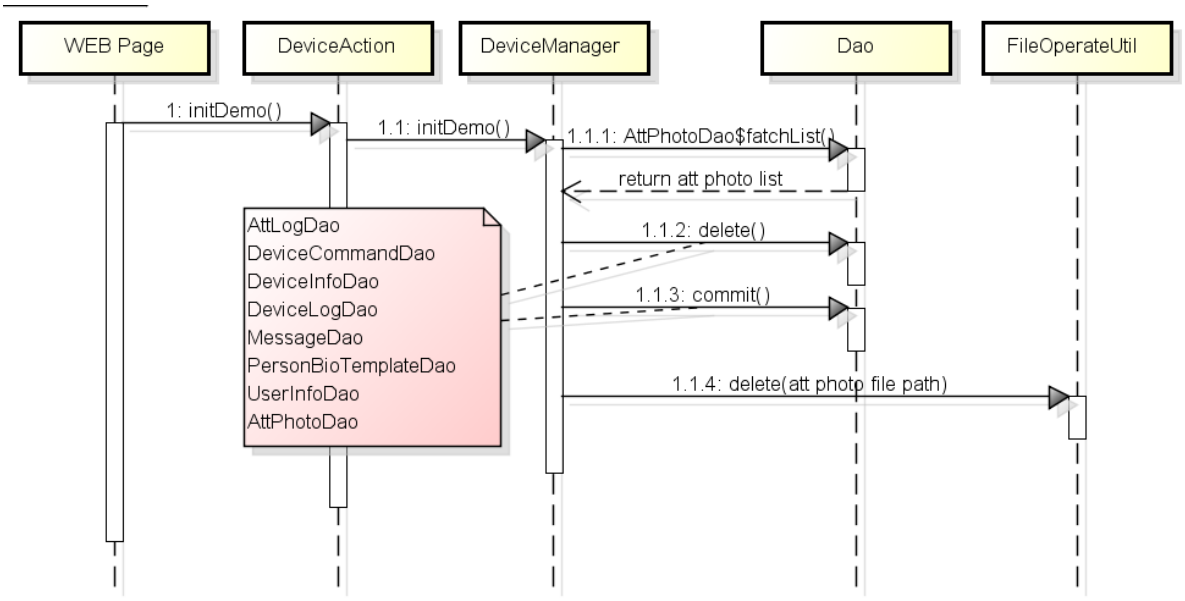
7.7.4 Deleting Executed Device Commands

This function deletes all the device commands that have been executed. Whether a device command has been executed depends on whether the **CMD_RETURN** value in the related database table is null.



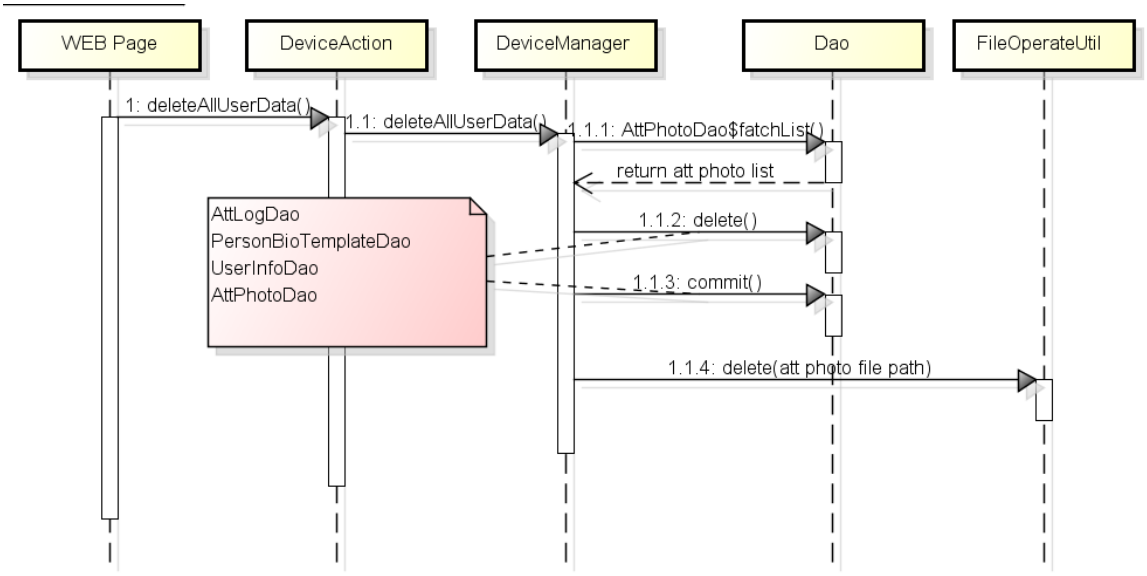
7.8 Initializing the Demo

This function deletes all database data and attendance photo files of the Demo.



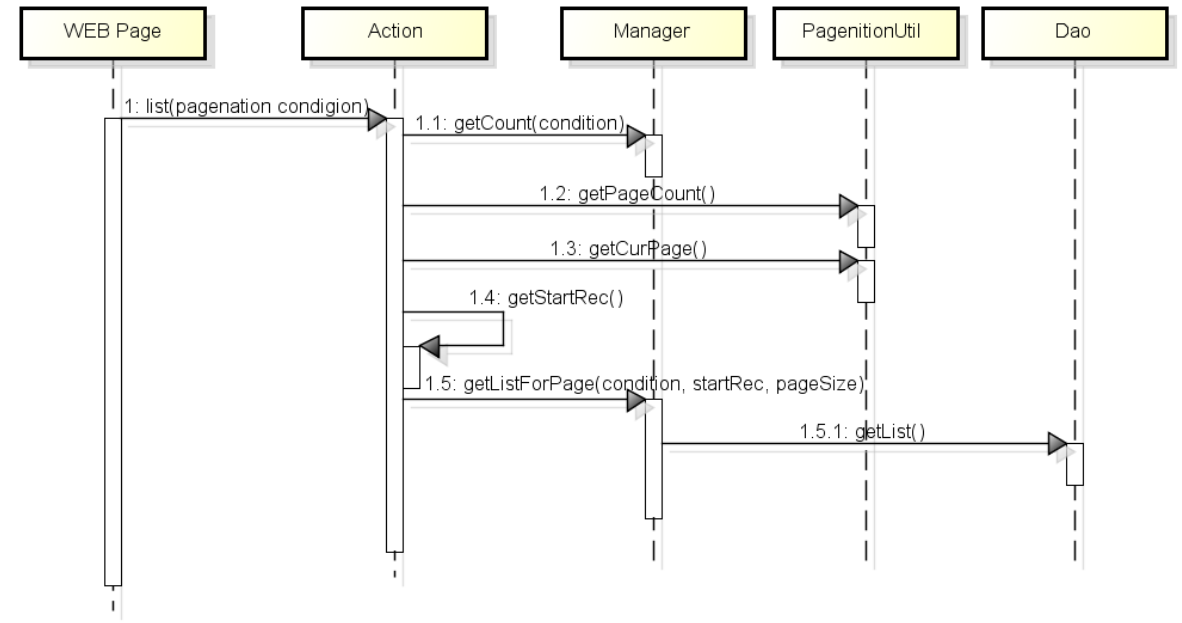
7.9 Deleting All User Data

This function deletes data of all users.



7.10 Displaying a List on Multiple Pages

In the list UI of the Demo, a list is displayed on multiple pages if it is lengthy. The Demo uses the method shown in the following figure to implement this function.


WEB Page
Action
Manager
PagenitionUtil
Dao
1: list(pagenation condigion)
1.1: getCount(condition)
1.2: getPageCount()
1.3: getCurPage()
1.4: getStartRec()
1.5: getListForPage(condition, startRec, pageSize)
1.5.1: getList()