THPV011

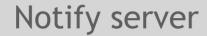
NOTIFICATIONS WITH NATIVE MOBILE APPLICATIONS

E. L

B. Bertrand*, J. Forsberg, MAX IV, Lund, Sweden







EUROPEAN

SOURCE

SPALLATION



REST API

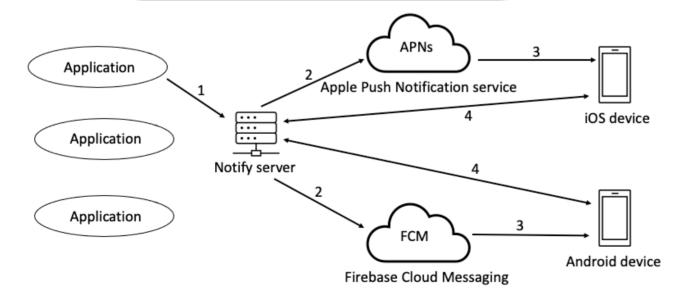
- Receive notifications
- Communicate with mobile clients

Sending notifications

POST to /services/{service_id}/notifications

A **service** is a category to group notifications.

Notification workflow



Notification workflow

iOS and Android clients



Messages list



ICALEPCS 2021 Demo Message Oct 1, 2021 at 11:00:3 Lorem ipsum dolor sit amet, consectetur adipiscing elit. Nam, ut paulo ante docui, augendae voluptatis finis est doloris omnis amotio. Hoc est vim afferre, Torquate, sensibus, extorquere ex animis cognitiones verborum, quibus inbuti sumus. Et ille ridens: Video, inquit, quid agas; At iste non dolendi status non vocatur voluptas. Quae cum essent dicta, discessimus. Re mihi non aeque satisfacit, et quidem locis pluribus. In quibus doctissimi illi veteres inesse quiddam caeleste et divinum putaverunt. Ergo id est convenienter naturae vivere, a natura discedere. Quod maxime efficit Theophrasti de beata vita liber, in quo multum admodum fortunae datur. Quid ait Aristoteles reliquique Platonis alumni? Bonum incolumis acies: misera caecitas. Ab his oratores, ab his imperatores ac rerum publicarum principes extiterunt.

Single message view

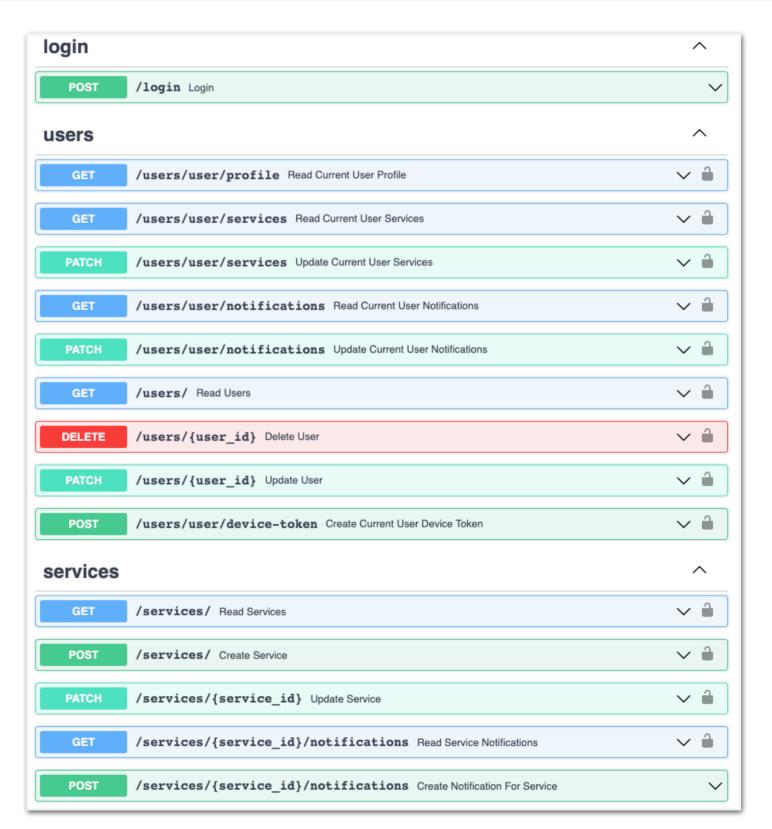


Notify Server



- Async Python web framework
- Based on Starlette, a lightweight ASGI framework and Pydantic, a data validation library using python type annotations
- High performance
- Fast to code
- Based on OpenAPI standard
- API documentation via Swagger UI
- Postgresql as database

```
@router.get("/", response_model=List[schemas.Service])
def read_services(
    db: Session = Depends(deps.get_db),
    current_user: models.User = Depends(deps.get_current_user),
):
    """Read all services"""
    db_services = crud.get_services(db)
    return db_services
```





Sending notifications



Send a **POST** to /services/{service_id}/notifications with the fields:

- title
- subtitle
- url

A **service** is a category to group notifications.

Each service is identified by a *service_id* (UUID).

Users can subscribe to the service they want.

```
curl -X 'POST' \
   'https://notify.maxiv.lu.se/api/v2/services/f848b451-e3c8-497b-a6da-2cb571d9d95e/notifications' \
   -H 'accept: application/json' \
   -H 'Content-Type: application/json' \
   -d '{
    "title": "My message",
    "subtitle": "This is a test",
    "url": ""
}'
```

Sending a notification with curl

```
data = {"title": title, "subtitle": subtitle, "url": url}
response = httpx.post(
    f"{server}/api/v1/services/{service_id}/notifications", json=data
)
```

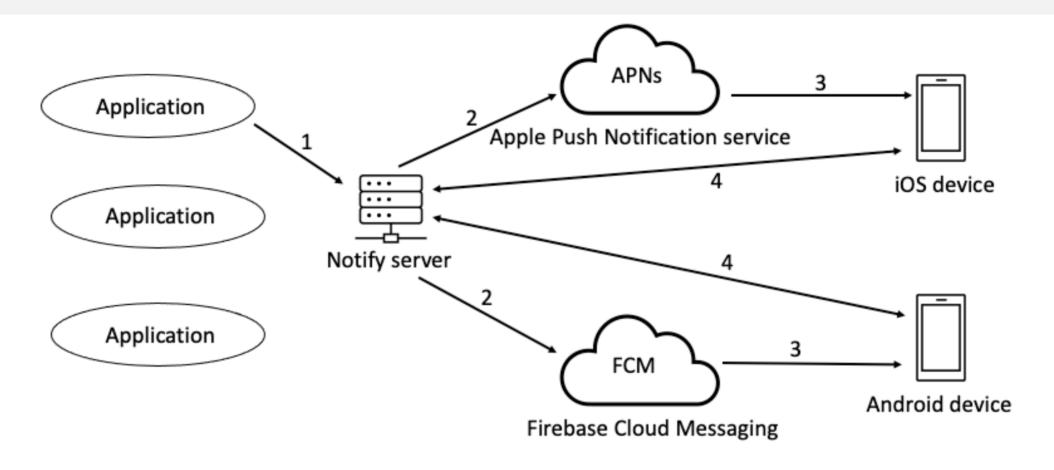
Integrated in:

- ESS LogBook
- OpenXAL
- EPICS alarms via Kafka
- Prometheus
- Achtung (new alarm management system for Tango)



Notification workflow





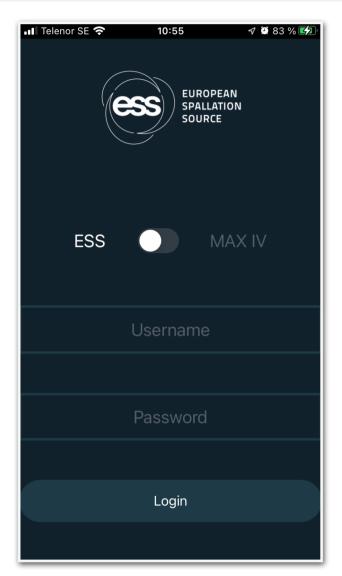
Notification workflow

- 1. An application sends a message to the Notify Server (the message is linked to a service).
- 2. The server sends a notification to both Apple Push Notification service and Firebase Cloud Messaging (depending on the device token type) for all users who subscribed to that service.
- 3. The notification is sent by Apple and Google cloud services to the user device.
- 4. When the user opens the notification, the full message is read from the Notify server API.

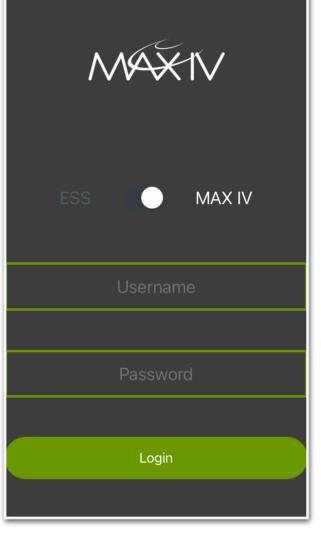


iOS and Android clients





ESS Login screen (iOS)

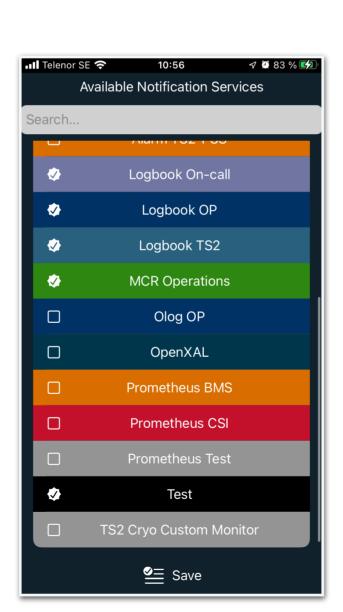


10:55

4 9 83 % **5**

■■ Telenor SE 🤝

MAX IV Login screen (iOS)





Push notification alert (Android)

