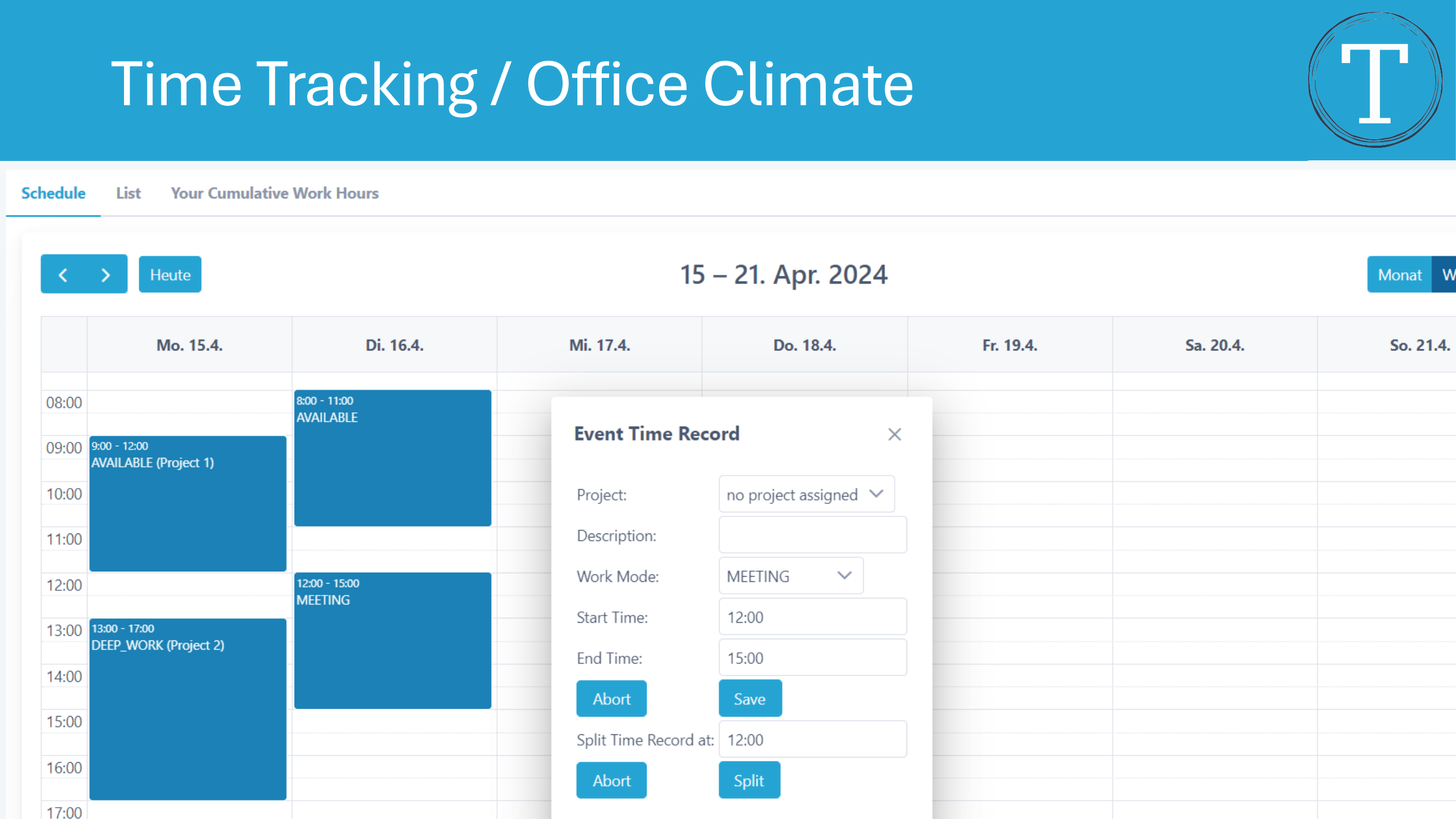
Projekt Tempera – Zwischenpräsentation

Jonas Bayer
Lea Hof
Manuela Niedrist
Astrid Reisinger
Niklas Speckle

Time Tracking / Office Climate



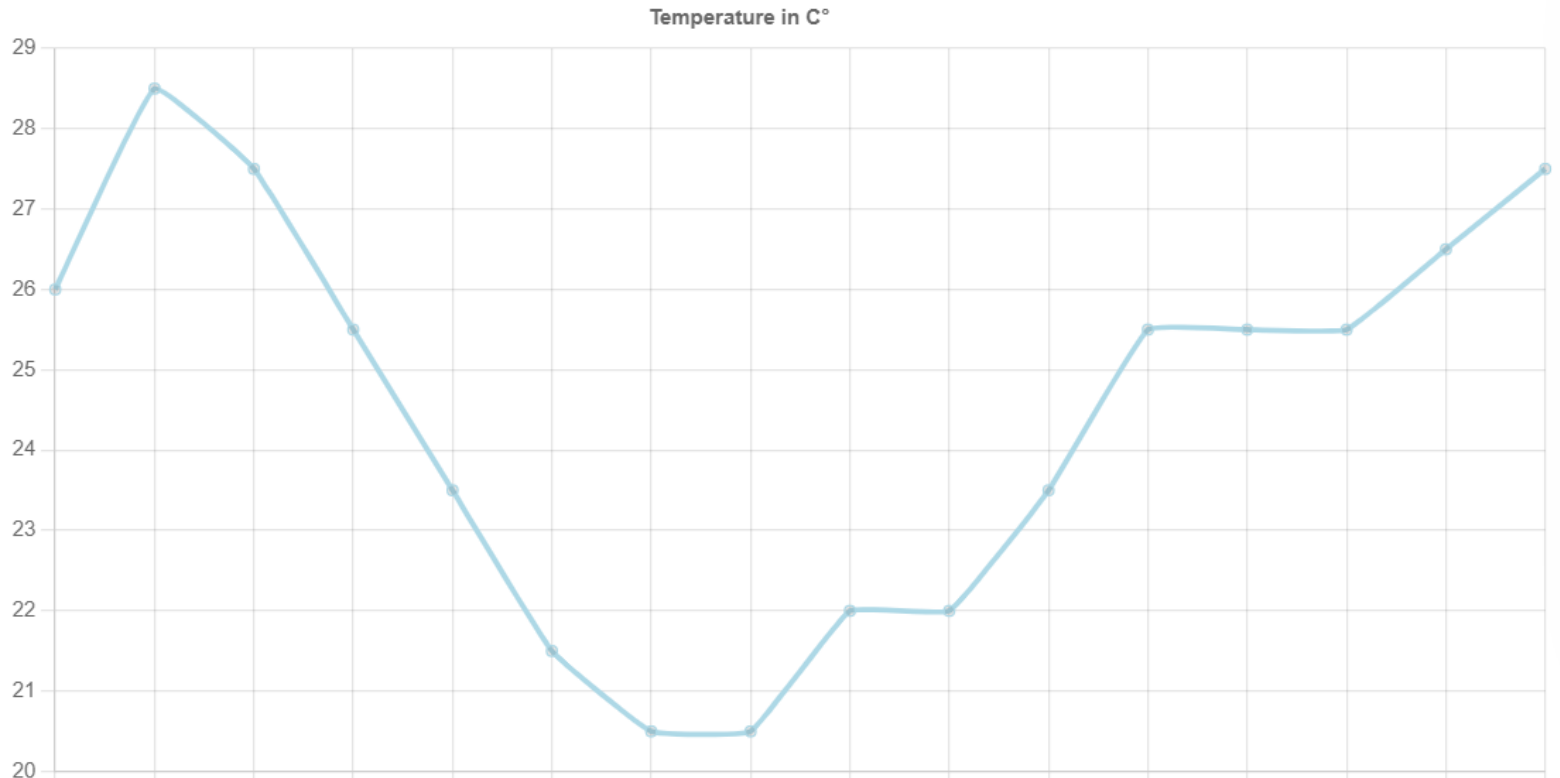
Time Tracking / Office Climate

Temperature	Humidity	Air Quality	Light
24.0 C°	40.0 %	30.0 ppm	690.0 lux

Temperature | Humidity | Air Quality | Light

Too warm: Check for open windows and doors and turn on ac
Too cold: Check for open windows and doors and turn up the heating.

Overview of past climate data:



Settings for Chart

datatype:

- Air Temperature
- Air Humidity
- Air Quality
- Light Intensity

Range: 03.05.24 - 04.05.24

granularity in minutes: 2

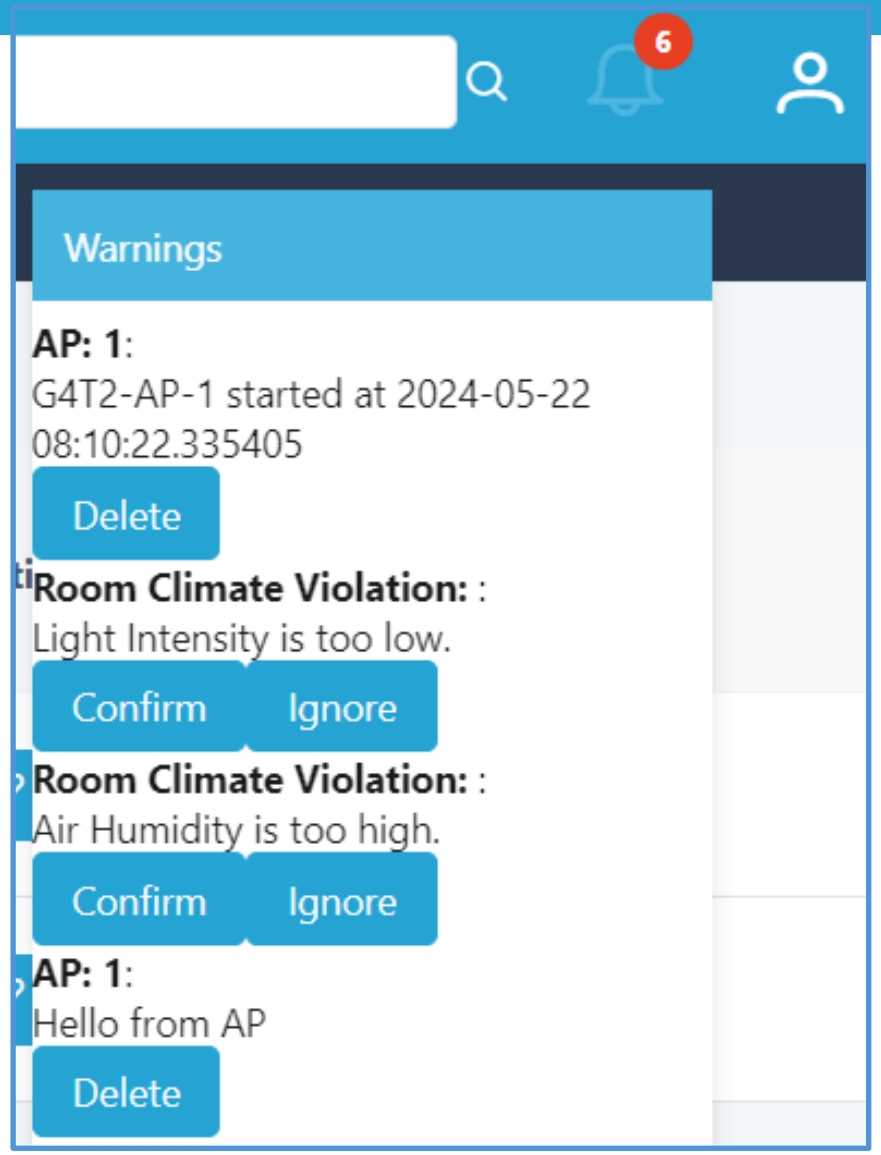
✓ Submit

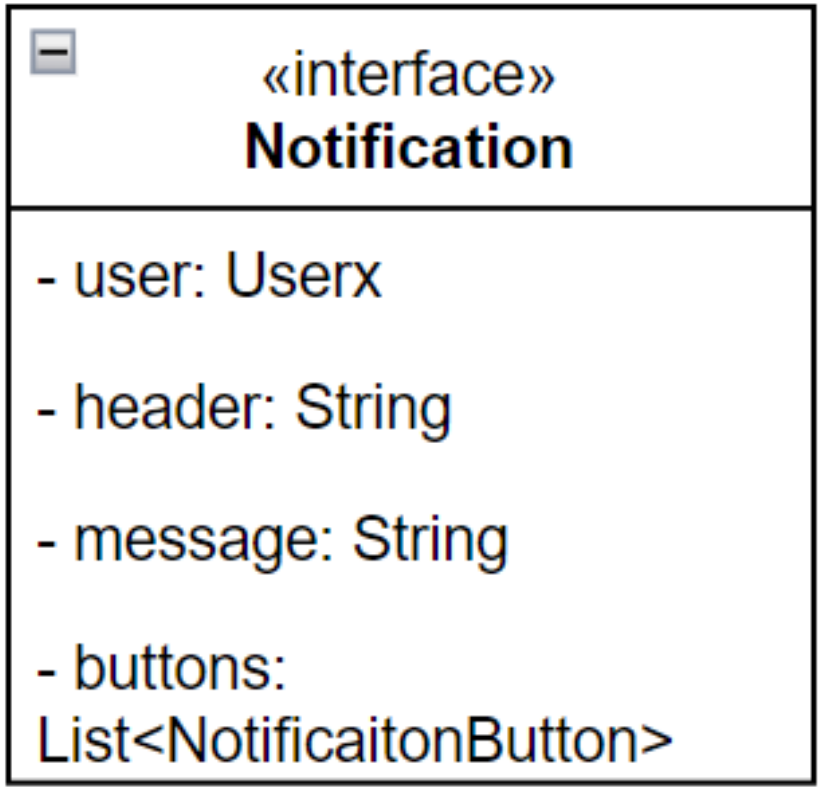
Feature: Notifications

- Durch Tempera Device bei fehlerhafter Verbindung

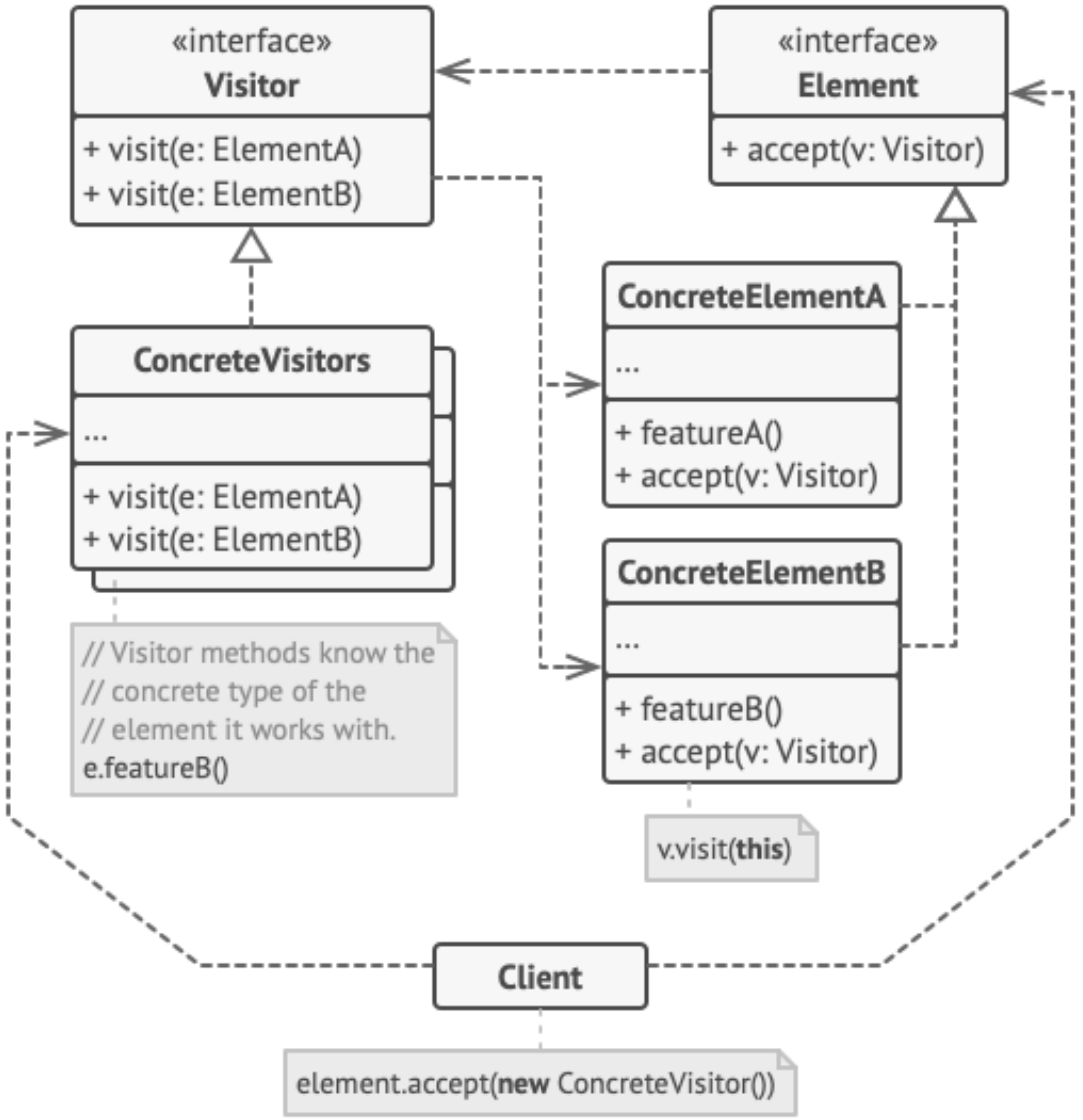
- Durch Access Point mittels POST-Request

Feature: Notifications





Visitor Pattern

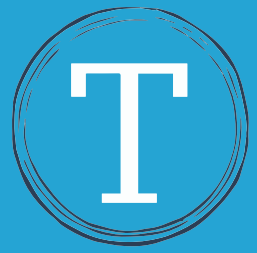


NotificationButton

- NotificationDeleteB
- NotificationConfirmB
- NotificationIgnoreB

Quelle: https://refactoring.guru/

Observer Pattern – Token

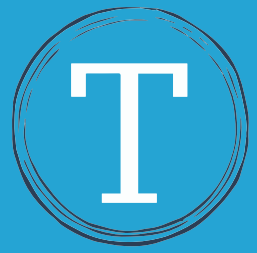


```
@Transactional  2 usages   csaf7520 *
public void updateWarningStatus(Warning warning,
                                WarningStatus warningStatus,
                                TemperaDevice temperaDevice) {
    warning.setWarningStatus(warningStatus);
    warningRepository.save(warning);

    tokenService.checkToken(warning);

    applicationEventPublisher.publishEvent(new NotificationEvent
            ( source: this, temperaDevice, warning));
}
```

Listener-Klassen:
- ListNotification
- EmailNotification

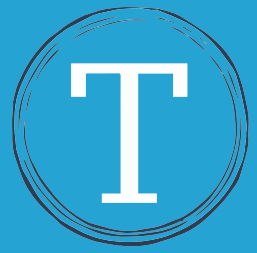
onApplicationEvent
(NotificationEvent event)

Feature: Logging als Aspect



- Aspect Oriented Programming
- Mittles Proxy
- Power Type Pattern: Entitätsklasse ←→ Metaklasse

Feature: Logging als Aspect



```java
@AfterReturning(pointcut="execution(* at.qe.skeleton.services.UserService.deleteUser(..))" +
        "&& args(user)", argNames= "user")
public void logDeleteUser(Userx user) {
    AuditLog log = new AuditLog();
    log.setAction(Action.DELETE);
    log.setStatus(ActionStatus.SUCCESS);
    log.setAccessedResource("USER: "+user.getUsername()+" (id: "+user.getId()+")");
    log.setAuthenticatedUser(getAuthenticatedUser());
    log.setTimestamp(LocalDateTime.now());
    logRepository.save(log);
}
```

Auszug Aspect-Klasse

```java
@Entity
@EntityListeners(AuditingEntityListener.class)
public class Userx extends Metadata
```

Power Type Pattern

Danke für eure Aufmerksamkeit!