

[Worker Chart] 접속자 데이터셋 API 연동 가이드 문서

작성자 : 김석현

개정이력

버전	변경일	변경사유	변경내역
V 1.0	2020-08-30	최초 작성	최초 작성
V 1.1	2020-09-04	휴일 URL 수정	/Average/notHoliday -> /average/not-holiday
V 2.0	2020-09-18	반환 값 변경	목차 수정, 모든 반환 값 수정, 변경

공통 데이터 목록

Name	Description	Type	Note
title	요청한 데이터 명	String	
description	요청한 데이터 설명	String	
_links.self	호출된 URL 반환	String	

응답은 다음과 같은 규칙을 따른다.

- 1. JSON 포맷의 데이터(Object Type)이며, 필드명은 명사만을 사용한다.

API 문서 목차

- 1. 회원 가입
- 2. 년도 기준 접속 요청
- 3. 년도, 월별 기준 접속 요청
- 4. 년도, 월, 일별 기준 접속 요청
- 5. 모든 기간의 평균 접속 요청
- 6. 휴일을 제외한 모든 기간의 평균 접속 요청
- 7. 년도, 월, 부서별 접속 요청
- 8. 모든 기간의 부서별 접속 요청
- 9. 모든 기간의 접속 요청 엑셀 파일 다운로드

회원가입

요청 데이터 필드

Parameter	Mandatory	Description
userID	O	계정 아이디
username	O	사용자 이름
hr_Organ	O	사용자의 부서 명 (영문)
password	O	계정 비밀번호

요청 방법

```
<form action="/api/account" method="POST">
  <div class="form-group">
    <label for="id" th:text="userID">아이디</label>
    <input type="text" name="id" id="id" placeholder="아이디를 입력하세요." minlength="2" maxlength="8"/>
    <label for="name" th:text="username">이름</label>
    <input type="text" name="name" id="name" placeholder="닉네임 입력하세요." minlength="5" maxlength="15"/>
    <label for="Organ" th:text="hr_Organ">부서명 (영문)</label>
    <input type="text" name="Organ" id="Organ" placeholder="부서명을 입력하세요." minlength="5"
maxlength="20"/>
    <label for="password" th:text="password">비밀번호</label>
    <input type="password" name="password" id="password" placeholder="비밀번호 입력하세요." minlength="8"
maxlength="50"/>
  </div>
  <button type="submit">유저 등록</button>
</form>
```

요청 구조

```
POST /api/account HTTP/1.1
Content-Type: application/json;charset=UTF-8
Content-Length:
Host: localhost:8080
{
  "userID" : "AAA",
  "username" : "test",
  "hr_Organ" : "planning Dept.",
  "password" : "1234255215"
}
```

년도 기준 접속 요청

Parameter	Mandatory	Description
year	O	검색할 Year (ex: “2020”)

요청 URL

GET /api/users/{year}
ex) let response = await fetch('http://localhost:8080/api/users/2020');

응답 구조

```
{
  "title": "2020년 접속자 통계",
  "description": "2020년 기준 접속자 통계입니다.",
  "year": "2020",
  "month": "ALL",
  "days": "ALL",
  "total_Count": "55",
  "_links": {
    "self": {
      "href": "http://localhost:8080/api/users/2020"
    },
    "prev-link": {
      "href": "http://localhost:8080/api/users"
    }
  }
}
```

년도, 월별 기준 접속 요청

요청 데이터 필드

Parameter	Mandatory	Description
id	O	로그인 USER ID (ex: “AAA”)
year	O	검색할 Month의 Year (ex: “2020”)
month	O	검색할 Month(ex: “09”)

요청 URL

GET /api/users/{id}/{year}/{month}
ex) let response = await fetch('http://localhost:8080/api/users/AAA/2020/09');

응답 구조

<pre>{ "title": "2020년 접속 요청 통계", "description": "2020년 9월 기준 접속 요청 통계입니다.", "year": "2020", "month": "09", "days": "ALL", "total_Count": "49", "_links": { "self": { "href": "http://localhost:8080/api/users/AAA/2020/09" }, "prev-link": { "href": "http://localhost:8080" } } }</pre>

Name	Description	Type	Note
total_Count	년월 기준 접속자 수	String	
_links.prev-link	이전 페이지 URI	String	

년도, 월, 일별 기준 접속 요청

요청 데이터 필드

Parameter	Mandatory	Description
id	O	로그인 USER ID (ex: "AAA")
year	O	검색할 Month의 Year (ex: "2020")
month	O	검색할 Month(ex: "09")
day	O	검색할 Day(ex: "17")

요청 URL

GET /api/users/{id}/{year}/{month}/{day}
ex) let response = await fetch('http://localhost:8080/api/users/AAA/2020/09/17');

응답 구조

```
{
  "title": "2020년 09월 17일 기준 요청 통계",
  "description": "2020년 9월 17일 기준 접속 요청 통계입니다.",
  "year": "2020",
  "month": "09",
  "days": "17",
  "total_Count": "2",
  "_links": {
    "self": {
      "href": "http://localhost:8080/api/users/2020/09/17"
    },
    "prev-link": {
      "href": "http://localhost:8080"
    }
  }
}
```

Name	Description	Type	Note
total_Count	년월일 기준 접속자 통계	String	
_links.prev-link	이전 페이지 URI	String	

모든 기간의 평균 접속 요청

요청 데이터 필드

Parameter	Mandatory	Description
id	O	로그인 USER ID (ex: “AAA”)

요청 URL

GET /api/users/{id}/day/Average

ex) let response = await fetch('http://localhost:8080/api/users/AAA/day/average');

응답 구조

<pre>{ "title": "모든 기간 접속 요청 통계", "description": "모든 기간 접속 요청 통계입니다.", "year": "ALL", "month": "ALL", "days": "ALL", "total_Count": "6.555556", "_links": { "self": { "href": "http://localhost:8080/api/users/AAA/day/average" }, "prev-link": { "href": "http://localhost:8080" } } }</pre>

Name	Description	Type	Note
total_Count	평균 하루 로그인 수 통계	String	
_links.prev-link	이전 페이지 URI	String	

휴일을 제외한 모든 기간의 평균 접속 요청

요청 데이터 필드

Parameter	Mandatory	Description
id	O	로그인 USER ID (ex: “AAA”)

요청 URL

GET /api/users/{id}/day/average/not-holiday
ex) let response = await fetch('http://localhost/api/users/1/day/average/not-holiday');

응답 구조

<pre>{ "title": "휴일을 제외한 모든 기간 접속 요청 통계", "description": "휴일을 제외한 모든 기간 접속 요청 통계입니다.", "year": "ALL", "month": "ALL", "days": "Not Holiday", "total_Count": "8.285714", "_links": { "self": { "href": "http://localhost:8080/api/users/AAA/day/average/not-holiday" }, "prev-link": { "href": "http://localhost:8080" } } }</pre>

Name	Description	Type	Note
total_Count	휴일 제외 로그인 수 통계	String	
_links.prev-link	이전 페이지 URI	String	

년도, 월, 부서별 접속 요청

요청 데이터 필드

Parameter	Mandatory	Description
id	O	로그인 USER ID (ex: “AAA”)
Organ	O	검색할 부서의 이름 (ex: “planning team”)
year	O	검색할 Month의 Year (ex: “2020”)
month	O	검색할 Month(ex: “09”)

요청 URL

```
GET /api/users/{id}/{year}/{month}/{Organ}
ex let response = await fetch('http://localhost:8080/api/users/AAA/2020/09/dept/planning%20team');
```

응답 구조

```
{
  "title": "2020년 09월 planning team부서 접속 요청 통계",
  "description": "2020년 9월 planning team부서 기준 요청 통계입니다.",
  "year": "2020",
  "month": "09",
  "days": "ALL",
  "total_Count": "43",
  "_links": {
    "self": {
      "href": "http://localhost:8080/api/users/AAA/2020/09/dept/planning%20team"
    },
    "prev-link": {
      "href": "http://localhost:8080"
    }
  }
}
```

Name	Description	Type	Note
total_Count	년월일 기준 부서 접속자	String	
_links.prev-link	이전 페이지 URI	String	

모든 기간의 부서별 접속 요청

요청 데이터 필드

Parameter	Mandatory	Description
id	O	로그인 USER ID (ex: "AAA")
Organ	O	검색할 부서의 이름 (ex: "planning team")
year	O	검색할 Month의 Year (ex: "2020")
month	O	검색할 Month(ex: "09")

요청 URL

GET /api/users/{id}/{Organ}

ex let response = await fetch('http://localhost:8080/api/users/AAA/dept/planning%20team');

응답 구조

<pre>{ "title": "모든 기간 planning team부서 요청 통계", "description": "모든 기간 planning team부서 요청 통계입니다.", "year": "ALL", "month": "ALL", "days": "ALL", "total_Count": "50", "_links": { "self": { "href": "http://localhost:8080/api/users/AAA/dept/planning%20team" }, "prev-link": { "href": "http://localhost:8080" } } }</pre>
--

Name	Description	Type	Note
total_Count	년월일 기준 부서 접속자	String	
_links.prev-link	이전 페이지 URI	String	

모든 기간의 접속 요청 엑셀 파일 다운로드

요청

```
GET /api/downloadExcelFile
ex let response = await fetch('http://localhost:8080/api/downloadExcelFile')
```

응답 헤더

```
GET / HTTP/1.1
Host: localhost:8080
Connection: keep-alive
Cache-Control: max-age=0
Upgrade-Insecure-Requests: 1
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko)
Chrome/85.0.4183.102 Safari/537.36
Accept:
text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.9
Referer: http://localhost:8080/
Accept-Encoding: gzip, deflate, br
Accept-Language: ko-KR,ko;q=0.9,en-US;q=0.8,en;q=0.7
```

Name	Description	Type	Note
downloadExcelFile.xlsx	API의 모든 접속 요청을 엑셀로 출력	.xlsx	