

2022

**TEAM Go Borrow!**

**Lends PROJECT**

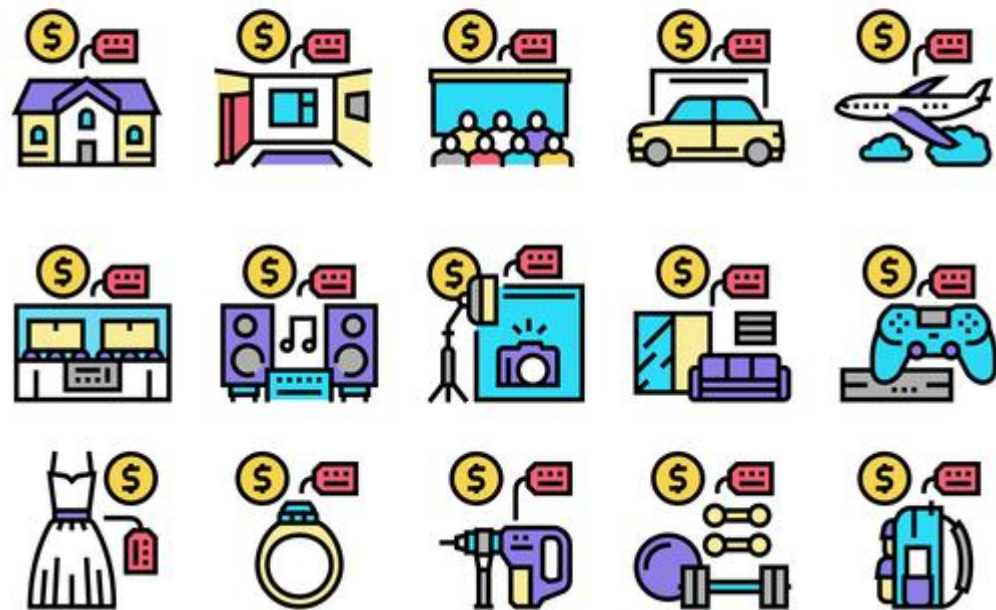
이성준 윤준성 최한운 최지윤

---


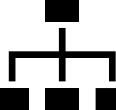

제2차 발표 (2022. 03. 31)

# 프로젝트 개요

## RENTAL



## Go Borrow ?

- 대여 서비스 시장이 매우 커짐에 따라 사용자 
- 사용자가 대여를 편히 할 수 있는 서비스
- 기관 / 조직이 물품 관리를 쉽고 빠르게 관리 
- 자신이 원하는 물품과 대여 기간을 정해서 빌릴 수 있음 

## 기존 서비스와의 차별성 ?

- 여러 회사와 자신이 원하는 물품과 대여 기간을 정해서 빌릴 수 있음
- 학교와 공공기관에서의 대여 및 소속 인증 연결



# CONTENTS

1. 소프트웨어 개발론
2. 프로젝트 개발 환경
3. **Client** 개발 현황
4. **Server** 개발 현황
5. 향후 개발 일정



01

# 소프트웨어 개발론

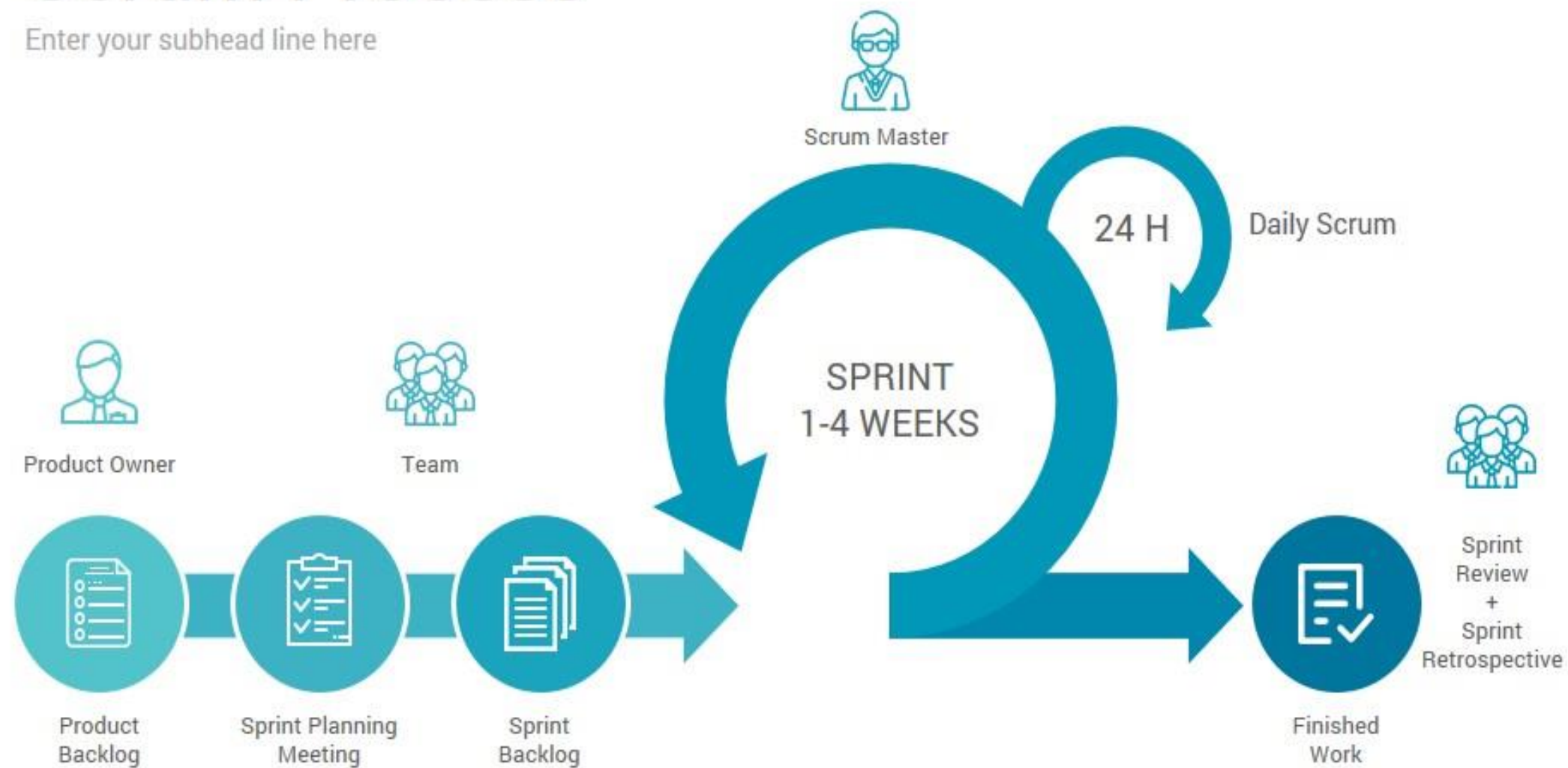
---

# 01. 소프트웨어 개발론

## Agile Scrum

### Scrum Process

Enter your subhead line here



Product Owner  
윤준성 [@wnstjd9701](#)



Scrum Master  
이성준 [@castlejun-2](#)



Development Team  
최한윤 [@chlgksdbs](#)



Development Team  
최지윤 [@J1Yun](#)

# 01. 소프트웨어 개발론

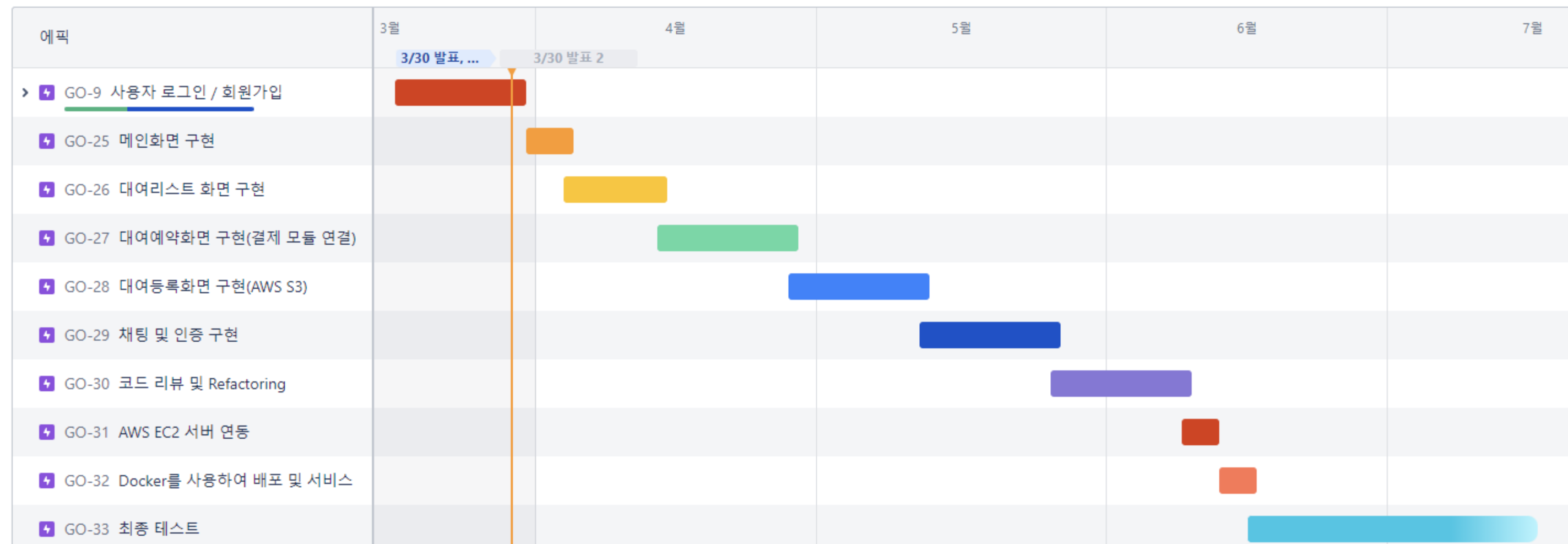
## Agile Scrum

프로젝트 / GoBorrow

로드맵

🔊 피드백 보내기

🔍  상태 범주 ▾



# 01. 소프트웨어 개발론

## Agile Scrum

🔑 사용자 로그인 / 회원가입 / 📄 GO-11

### 로그인 API

📎 첨부   🧑 하위 이슈 추가   🔗 이슈 연결   ▾   ⋮

#### 설명

사용자는 로그인을 위해

아이디와 패스워드를 입력하고

로그인 유지를 원한다.

#### 하위 이슈

을(를) 기준으로 정렬 ▾   ⋮   +

0% 완료

📄 GO-34	아이디와 패스워드가 일치하는지 확인하는 쿼리 작성	-	👤	해야 할 일 ▾
📄 GO-35	아이디 패스워드가 데이터베이스에 있으면 로그인 성공	-	👤	해야 할 일 ▾
📄 GO-36	아이디 패스워드 예외처리	-	👤	해야 할 일 ▾
📄 GO-40	로그인 성공 시 액세스 토큰과 리프레쉬 토큰을 발급	-	👤	해야 할 일 ▾

#### 활동

표시: 모두   **댓글**   기록

최신 항목 먼저 ↕



댓글 추가...

프로 팁: 눌러서 **M** 댓글 추가

🔒   👁 1   🍌   🔗   ⋮   ✕

진행 중 ▾

#### 세부 정보

담당자	👤 윤준성
레이블	없음
Sprint	사용자 로그인 / 회원가입
Story point estimate	없음
개발	🔗 브랜치 만들기 ▾
보고자	👤 윤준성

만들기 1시간 전  
업데이트됨 34분 전

⚙ 구성

# 01. 소프트웨어 개발론

## Agile Scrum

GO-11 / GO-40

1 1b 2 3 4 5

### 로그인 성공 시 액세스 토큰과 리프레시 토큰을 발급

첨부 이슈 연결

#### 설명

로그인 요청이 들어오면 JWT 를 이용하여 AccessToken과 RefreshToken을 발급 해준다.

AccessToken은 사용자가 로그인을 유지 시켜주고 기간이 만료되면 다시 로그인을 하게 된다.

RefreshToken은 AccessToken을 재발급하기 위한 토큰이다.

AccessToken이 만료되면 RefreshToken을 통해 재발급을 받게된다.

#### 활동

표시: 모두 댓글 기록

최신 항목 먼저



댓글 추가...

프로 팁: 눌러서 댓글 추가

해야 할 일

#### 세부 정보

담당자	윤준성
레이블	없음
Sprint	사용자 로그인 / 회원가입
Story point estimate	없음
개발	브랜치 만들기
보고자	윤준성

만들기 16분 전  
업데이트됨 16분 전

구성



# 01. 소프트웨어 개발론

## Agile Scrum

프로젝트 / GoBorrow

백로그

🔍 👤 + 에픽 ▾

📈 인사이트

에픽

에픽이(가) 없는 이슈

- > 사용자 로그인 / 회원가입
- > 메인화면 구현
- > 대여리스트 화면 구현
- > 대여예약화면 구현(결제 모듈 연결)
- > 대여등록화면 구현(AWS S3)
- > 채팅 및 인증 구현
- > 코드 리뷰 및 Refactoring
- > AWS EC2 서버 연동
- > Docker를 사용하여 배포 및 서비스
- > 최종 테스트

+ 만들기 에픽

▼ 사용자 로그인 / 회원가입 17 3월 - 30 3월 (4개 이슈) 스프린트 완료

로그인 / 회원가입 구현 JWT 미들웨어 / JWT토큰 인증 구현

GO-11 로그인 API	사용자 로그인 / 회원가입	진행 중	👤
GO-12 회원가입 API	사용자 로그인 / 회원가입	진행 중	👤
GO-14 JWT 토큰 인증 API	사용자 로그인 / 회원가입	완료됨	👤
GO-13 JWT 미들웨어	사용자 로그인 / 회원가입	완료됨	👤

+ 이슈 만들기

▼ 백로그 (2개 이슈) 스프린트 만들기

GO-19 로그인 페이지	사용자 로그인 / 회원가입	진행 중	👤
GO-20 회원가입 페이지	사용자 로그인 / 회원가입	진행 중	👤

+ 이슈 만들기

💡 Quickstart ✕

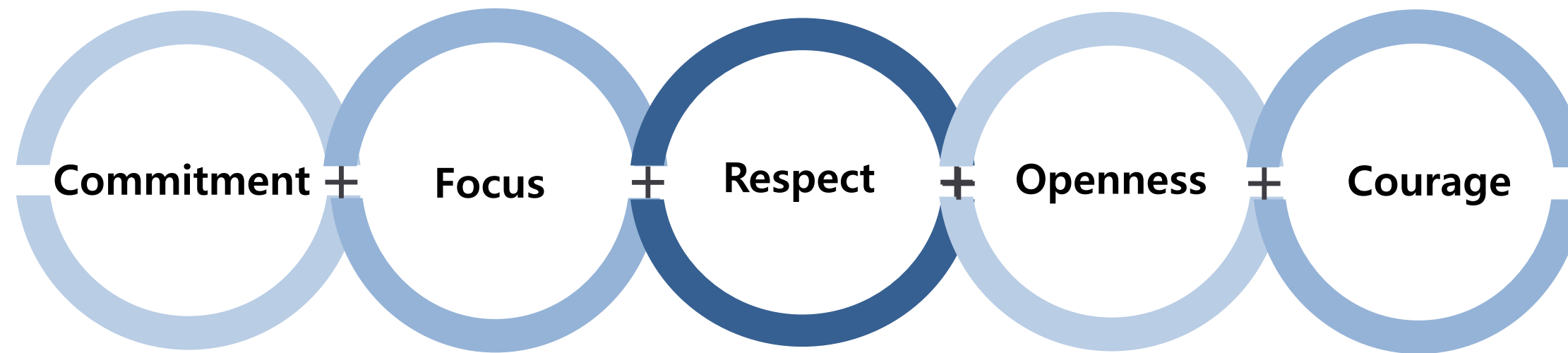
02

## 프로젝트 개발 환경

---


## 02. 프로젝트 개발 환경

### Five Team Keywords



## 02. 프로젝트 개발 환경

### 협업 환경

 **Notion** - 체크리스트, 회의록

#### GoBorrow 회의록

📅 리스트 보기 📊 표 + 보기 추가

필터 정렬 🔍 ... 새로 만들기 ▼

📅 생성일 ▼ 👤 생성인 ▼ + 필터 추가

📄 2022-03-21

2022년 3월 21일

📄 2022-03-22

2022년 3월 22일

📄 2022-03-23

2022년 3월 23일

📄 2022-03-24

2022년 3월 24일

📄 2022-03-25

2022년 3월 25일

📄 2022-03-28

2022년 3월 28일

📄 2022-03-29

2022년 3월 29일

📄 2022-03-30

2022년 3월 30일

+ 새로 만들기

## 02. 프로젝트 개발 환경

# 협업 환경



Google Docs

# Google Docs - API 명세서

API 명세서									
					RESTful API				
		API Index							
			Index	Method	URI	Description	명세서	서버 반영 여부	사용자 미들웨어
				POST	/app/users/login	로그인 API	ok	ok	
				GET	/app/auth/token	토큰 재발급 API	ok		
				POST	/app/users	사용자 회원가입	ok	ok	
				PUT	/app/users/:userId	사용자 정보수정 API	ok		
				POST	/app/users/:userId/organizations/favorites	조직/기관 즐겨찾기 등록 API	ok		
				PUT	/app/users/:userId/organizations/favorites	조직/기관 즐겨찾기 수정 API	ok		
				POST	/app/users/:userId/borrow/products/:productId	사용자 대여 신청 API	ok		
				GET	/app/users/:userId/borrow/products	사용자가 대여한 모든 품목 검색 API	ok		
				GET	/app/users/:userId/borrow/products/:productId	사용자가 대여한 특정 품목 검색 API	ok		
				GET	/app/products	대여 품목 검색 API	ok		
				POST	/app/products	대여 품목 등록 API	ok		
				PUT	/app/product/:productId	대여 품목 수정 API	ok		
				PUT	/app/products/lend/days	대여 가능일 설정 API	ok		
				GET	/app/products/:productId	대여 품목별 신청/사용 현황 API	ok		
				PUT	/app/products/:productId	대여 물품 수정 API	ok		middleware
				GET	/app/organizations	전체 조직/기관 조회 API (+검색)	ok		

/app/users/login

사용자 로그인 API

API 인덱스

2

Method

POST

Header

Key

Values

Body

Name

Type

Mandatory

Example

Default

Description

id

String

Y

test1

사용자 아이디

password

String

Y

1234

비밀번호

distinction

String

Y

organization

조직/ 사용자 구별

Query String

Name

Type

Mandatory

Example

Default

Description

Response Parameters

Name

Type

Mandatory

Example

Default

Description

isSuccess

boolean

Y

TRUE

요청 성공 여부

code

int

Y

1000

응답 코드

message

String

Y

성공

응답 메시지

accessToken

String

Y

...

access 토큰

refreshToken

String

Y

...

refresh 토큰

Response Sample

POST

localhost:5000/app/users/login

Params

Authorization

Headers (9)

Body ●

Pre-request Script

Tests

Settings

● none

● form-data

● x-www-form-urlencoded

● raw

● binary

● GraphQL

JSON ▼

1 {

2 "id": "test12",

3 "password": "1234",

4 "distinction": "1"

5 }

Body

Cookies (2)

Headers (9)

Test Results

Pretty

Raw

Preview

Visualize

JSON ▼

1 {

2 "password": "f



## 02. 프로젝트 개발 환경

### 협업 환경

#### Folder Structure

```
client
├── public
│   ├── index.html
│   ├── manifest.json
│   └── robots.txt
├── src
│   ├── _actions
│   │   ├── action_type.js
│   │   └── user_action.js
│   ├── _reducers
│   │   ├── index.js
│   │   └── user_reducer.js
│   ├── components
│   │   └── views
│   │       ├── Footer
│   │       │   ├── Footer.js
│   │       ├── Header
│   │       │   ├── Header.js
│   │       ├── LandingPage
│   │       │   ├── LandingPage.js
│   │       └── LoginPage
│   │           ├── LoginLandingPage.js
│   │           ├── LoginPage.js
│   │           └── RegisterPage.js
├── App.css
├── App.js
├── index.css
├── index.js
├── logo.svg
├── reportWebVitals.js
├── .gitignore
├── package-lock.json
└── package.json
```



#### Github - 코드리뷰, 코드 관리

```
server
├── config
│   ├── baseResponseStatus.js
│   ├── database.js
│   ├── express.js
│   ├── jwtMiddleware.js
│   └── response.js
├── src
│   ├── app
│   │   ├── User
│   │   │   ├── userController.js
│   │   │   ├── userDao.js
│   │   │   ├── userProvider.js
│   │   │   ├── userRoute.js
│   │   │   └── userService.js
│   │   └── Product
│   │       ├── productController.js
│   │       ├── productDao.js
│   │       ├── productProvider.js
│   │       ├── productRoute.js
│   │       └── productService.js
├── .gitignore
├── app.js
├── server.js
├── package.json
├── package-lock.json
└── README.md
```

master 2 branches 0 tags

Go to file

Add file

Code

wnstjd9701 Update: AccessToken Reissuance

bb26ca0 2 hours ago 53 commits

.github	Update: README.md	5 days ago
client	Lends Project	4 hours ago
server	Update: AccessToken Reissuance	2 hours ago
.babelrc	Lends Project	3 days ago
.gitignore	Lends Project	17 hours ago
README.md	Update: README.md	3 days ago
package.json	Lends Project	17 hours ago
yarn.lock	Lends Project	17 hours ago

03

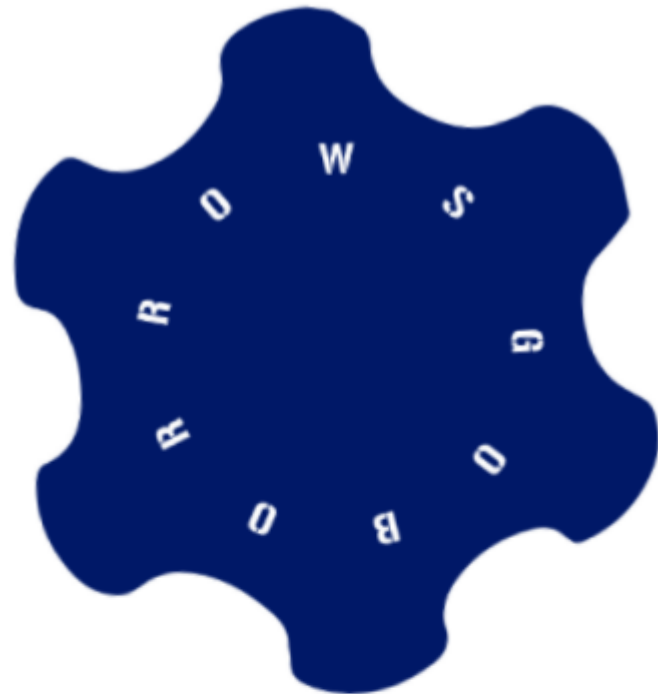
## Client 개발 진행 상황

---

## 03. Client 개발 현황

### 로그인 페이지

일반 사용자 회원가입

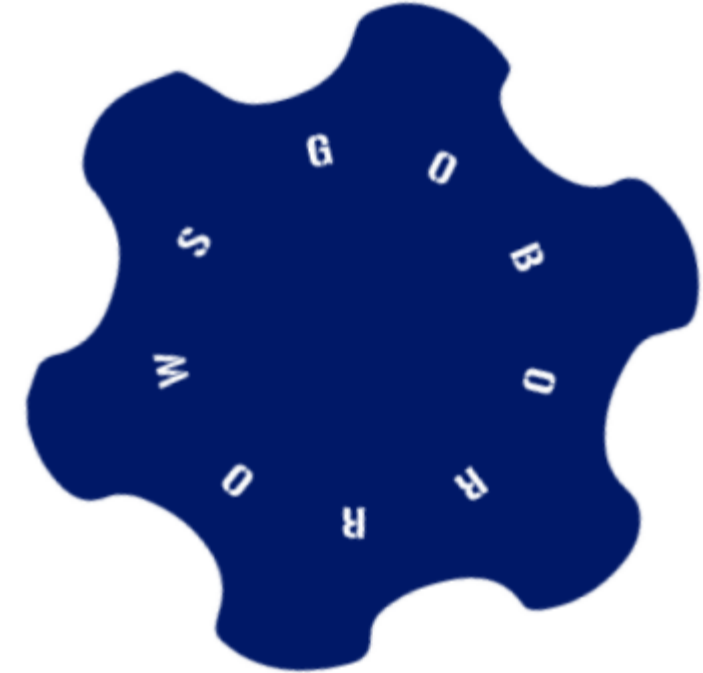


로그인(개인)	로그인(기업)
<input type="text" value="아이디"/>	
<input type="password" value="비밀번호"/>	
<input type="button" value="로그인"/>	

회원가입

©Dankook University's Team GoBorrow.

조직/기관  
사용자 회원가입



로그인(개인)	로그인(기업)
<input type="text" value="아이디"/>	
<input type="password" value="비밀번호"/>	
<input type="button" value="로그인"/>	

회원가입

©Dankook University's Team GoBorrow.

## 03. Client 개발 현황

### 회원가입 페이지

회원가입(개인)	회원가입(기업)
아이디	
비밀번호	
비밀번호 확인	
이름	
핸드폰 번호	
주소	
소개	
가입완료	

©Dankook University's Team GoBorrow.

회원가입(개인)	회원가입(기업)
아이디	
비밀번호	
비밀번호 확인	
기업 이름	
기업 대표자명	
대표 번호	
주소	
상세 주소	
기업 소개	
가입완료	

©Dankook University's Team GoBorrow.

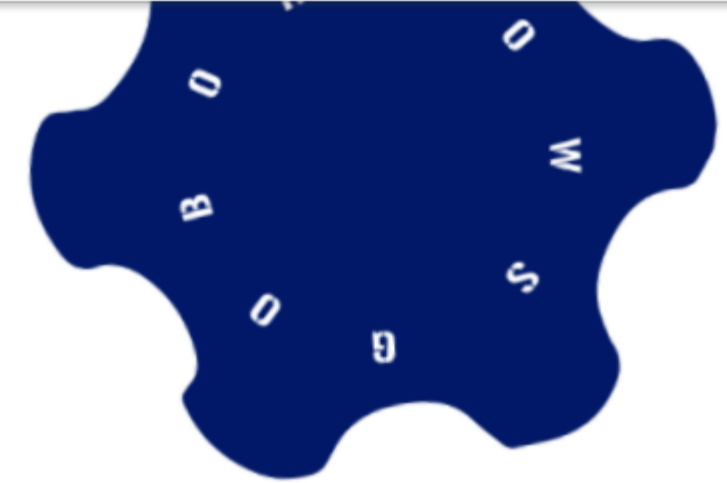


회원가입(개인)	회원가입(기업)
CapstoneProject	
.....	
.....	
GoBorrow	
010-2657-0707	
경기 용인시 수지구 죽전로 152	
단국대학교 소프트웨어학과 귀바로우 팀	
가입완료	

©Dankook University's Team GoBorrow.

localhost:3000 내용:  
회원가입에 성공하였습니다.

확인



04

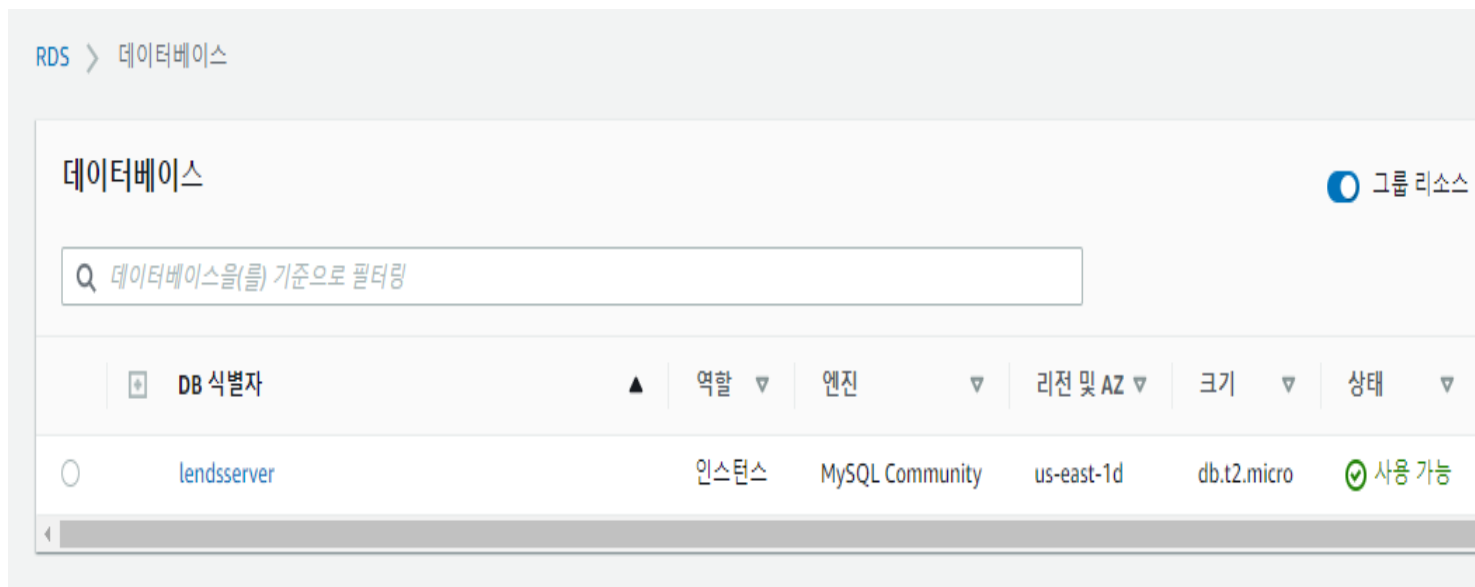
## Server 개발 현황

---



# 04. Server 개발 현황

## ERD 설계



User									
4		전화번호	phoneNumber	VARCHAR(20)					
5		주소	address	VARCHAR(45)					
6		구분	type	INT	1			user: 1, organization: 2	
7		정보	info	TEXT					
8		삭제여부	isDeleted	INT					
9		생성일	createdAt	DATE TIME	CURRENT_TIMESTAMP*				
10		수정일	updatedAt	DATE TIME	CURRENT_TIMESTAMP* ON UPDATE CURRENT_TIMESTAMP*				

Organization									
4		상세 주소	detaAddress	VARCHAR(45)					
5		기관 이름	organizationName	VARCHAR(45)					
6		담당자 이름	managerName	VARCHAR(20)					
7		전화번호	phoneNumber	VARCHAR(20)					
8		구분	type	INT	2			user: 1, organization: 2	
9		정보	info	TEXT					
10		삭제여부	isDeleted	INT					
11		생성일	createdAt	DATE TIME	CURRENT_TIMESTAMP*				
12		수정일	updatedAt	DATE TIME	CURRENT_TIMESTAMP* ON UPDATE CURRENT_TIMESTAMP*				

Category									
4				VARCHAR(45)					

Certification									
4		인증서 이미지 url	imageUrl	VARCHAR(200)					

Product									
4		대여품목 이름	productName	VARCHAR(45)					
5		보유 수량	quantity	INT					
6		대여품목 이미지 url	imageUrl	VARCHAR(300)					
7		품목 소개 / 대여 설명	info	TEXT					
8		대여 가격	price	INT					

Rent									
4		대여 상품 번호	itemId	INT					
5		대여 품목 번호	productId	INT					
6		대여 시작일	startDate	DATE TIME					
7		대여 종료일	endDate	DATE TIME					
8		대여 상태	rentStatus	VARCHAR(10)	WAITING			WAITING: 신청, CONF: 확정	
9		반납 일시	returnDate	DATE TIME					
10		반납 관련 비고	returnInfo	TEXT					

Item									
4		상품 상태	itemStatus	VARCHAR(50)					

Status									
4				VARCHAR(45)					

Bookmark									
4				VARCHAR(45)					

# 04. Server 개발 현황

## 회원가입 API

/app/users	사용자 회원가입 API	API 인덱스					
Method	POST	1					
Header	Name	Type	Mandatory	Example	Default	Description	
Body	Name	Type	Mandatory	Example	Default	Description	
	id	String	Y	test1		사용자 아이디	
	password	String	Y	1234		비밀번호	
	name	String	Y	jun		사용자 이름	
	phoneNumber	String	Y	010-1234-1234		휴대폰 번호	
	address	String	Y	경기도 용인시 수지구 죽전로		사용자 주소	
	info	String	Y	테스트 정보 입니다.		사용자 정보	
	distinction	String	Y	user or organization		조직/ 사용자 구별	
Query String	Name	Type	Mandatory	Example	Default	Description	
Response Parameters	Name	Type	Mandatory	Example	Default	Description	
	isSuccess	boolean	Y	TRUE		요청 성공 여부	
	code	int	Y	1000		응답 코드	
	message	String	Y	성공		응답 메시지	
Response Sample	<div><div>POSTlocalhost:5000/app/users</div><div>ParamsAuthorizationHeaders (8)BodyPre-request ScriptTestsSettings</div><div>● none● form-data● x-www-form-urlencoded● raw● binary● GraphQLJSON</div><div>1</div></div>						

```
/**
 * API No. 2
 * API Name : 사용자 로그인 API
 * [POST] /app/users/Login
 */
// type 1: 사용자, 2: 조직/기관
login = async function (req, res) {
  const type = parseInt(req.body.distinction); // req.body.type
  if (type === 1) {
    // 사용자 로그인
    const userId = req.body.id;
    const password = req.body.password;

    if (!userId) return res.send(NICKNAME_EMPTY); // code 2009
    if (userId.length > 20) return res.send(ID_LENGTH_ERROR); // code 2012
    if (!password) return res.send(PASSWORD_EMPTY); // code 2003
    if (password.length > 20) return res.send(PASSWORD_LENGTH_ERROR); // code 2013

    const loginResult = await userLogin(userId, password, type);
    res.cookie('refreshToken', loginResult.refreshToken, {
      httpOnly: true,
      maxAge: 3000000,
    });
    return res.send(loginResult);
  }
  // 조직/기관 로그인
  const organizationId = req.body.id;
  const password = req.body.password;

  if (!organizationId) return res.send(NICKNAME_EMPTY); // code 2009
  if (organizationId.length > 20) return res.send(ID_LENGTH_ERROR); // code 2012
  if (!password) return res.send(PASSWORD_EMPTY); // code 2003
  if (password.length > 20) return res.send(PASSWORD_LENGTH_ERROR); // code 2013

  const loginResult = await organizationUserLogin(organizationId, password, type);
  res.cookie('refreshToken', loginResult.refreshToken, {
    httpOnly: true,
    maxAge: 3000000,
  });
  return res.send(loginResult);
};

export default new userController();
```

## 04. Server 개발 현황

### 로그인 API

POST localhost:5000/app/users/login

Params Authorization Headers (10) **Body** Pre-request Script Tests Settings

● none ● form-data ● x-www-form-urlencoded ● raw ● binary ● GraphQL **JSON** ▾

```
1 {
2   ... "id": "test12",
3   ... "password": "1234",
4   ... "distinction": "1"
5 }
```

Body Cookies (2) Headers (9) Test Results

Pretty Raw Preview Visualize **JSON** ▾

```
1 {
2   "message": {
3     "isSuccess": true,
4     "code": 1000,
5     "message": "성공"
6   },
7   "accessToken": "eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJpZCI6InRlc3QxMiIsIm50IjoiMSJ9.eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJpZCI6InRlc3QxMiIsIm50IjoiMSJ9",
8   "refreshToken": "eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJpZCI6InRlc3QxMiIsIm50IjoiMSJ9"
9 }
```

```
const loginResult = await userLogin(id, password, distinction);
```

```
const userIdCheck = await idCheck(id);
if (userIdCheck.length < 1) return LOGIN_FAILURE; // code 1002 아이디가 존재 하지 않을 경우

const hashedPassword = createHash('sha512').update(password).digest('hex');
const params = [id, hashedPassword];
const checkResult = await getUserInfo(connection, params);
```

```
export async function selectUserId(connection, id) {
  const getUserId = `
    SELECT id FROM User WHERE id = ?
  `;
  const [userIdResult] = await connection.query(getUserId, id);
  return userIdResult;
}
```

```
return {
  message: SUCCESS,
  accessToken: accessToken,
  refreshToken: refreshToken,
};
```

## 04. Server 개발 현황

# JWT 토큰 재발급 API

/auth/token	JWT 토큰 재발급API	API 인덱스													
		3													
	Method	GET													
	Header	Key	Values												
		Authorization	Bearer AccessToken												
		refreshToken	refreshToken												
	Body	Name	Type	Mandatory	Example	Default	Description								
	Query String	Name	Type	Mandatory	Example	Default	Description								
	Response Parameters	Name	Type	Mandatory	Example	Default	Description								
		isSuccess	String	Y	TRUE										
		code	Integer	Y	1000										
		message	String	Y	성공										
		accessToken	String												
	Response Sample	<div><div>GETlocalhost:5000/auth/token</div><div><div>Params</div><div>Authorization</div><div>Headers (9)</div><div>Body</div><div>Pre-request Script</div><div>Tests</div><div>Settings</div></div><div><div>Headers</div><div>7 hidden</div></div><table><thead><tr><th>KEY</th><th>VALUE</th></tr></thead><tbody><tr><td><input checked="" type="checkbox"/> Authorization</td><td></td></tr><tr><td><input checked="" type="checkbox"/> refreshToken</td><td>eyJhbGciOiJIbGciLC</td></tr><tr><td>Key</td><td>Value</td></tr></tbody></table></div>						KEY	VALUE	<input checked="" type="checkbox"/> Authorization		<input checked="" type="checkbox"/> refreshToken	eyJhbGciOiJIbGciLC	Key	Value
KEY	VALUE														
<input checked="" type="checkbox"/> Authorization															
<input checked="" type="checkbox"/> refreshToken	eyJhbGciOiJIbGciLC														
Key	Value														

```
export const reissuanceAccessToken = async (req, res) => {  
  //const accessToken = req.headers.authorization.split('Bearer ')[1];  
  const refreshToken = req.headers.refreshtoken;
```

```
const refreshResult = await jwt.verify(refreshToken, process.env.JWT_SECRET, (err, data) => {
  if (err)
    return {
      isSuccess: false,
      code: 5002,
      name: err.name,
      message: err.message,
    };

  return {
    isSuccess: true,
    code: 5003,
    name: 'verify',
    message: 'Verify Success',
    id: data.id,
    distinction: data.distinction,
  };
});
```

```
return res.send({
  isSuccess: true,
  code: 5000,
  name: 'Reissuance',
  message: 'Reissuance NewAccessToken',
  accessToken: newAccessToken,
});
}
```

05

## 개발 일정

---



## 05. 개발 일정

### 향후개발 계획 일정

⚠ 2주 간격의 스프린트를 통해 추가적인 요구사항 반영 및 개발과정에서의 피드백 진행

~3.29	~4.4	~4.18	~5.2
<del>-서버 구축</del> <del>-로그인 및 회원가입</del> <del>(기관 및 개인 분류)</del>	-메인화면 구현	-대여리스트화면 구현 (위치 기반 Filter)	-대여예약화면 구현 (결제 모듈 연결)
~5.16	~5.30	~6.13	
-대여등록화면 구현 (AWS S3 활용)	-채팅 및 인증 구현 (Socket IO 활용)	-코드 Refactoring -AWS EC2 서버 연결	-시연영상 준비 -최종 발표 자료 준비

**Thank you**