

소프트웨어 마에스트로 3기 - 1차 프로젝트 -

할로윈 카니발 -ToT-

서버 API 1.1 Document

오 혜성

2012-09-04

소프트웨어 마에스트로 3기 - SNG 프로토 타입 제작 -(담당 멘토님: 김 성용)을 진행하면서 제작한 간단한 서버 API 사용에 관련된 기술 문서입니다.

목차

1. 시작하기에 앞서	5
1.1 이 문서에 대하여	5
1.2 개발 환경	5
1.3 API 테스트 페이지	5
1.4 관련 자료 다운로드	6
2. 회원가입 모듈	7
2.1 게임에서의 회원 가입 진행 방식	7
2.2 기본 회원 가입 UI	7
2.3 회원 가입	8
2.3.1 이전 회원 가입 스타일	8
2.3.2 변경된 회원 가입 스타일	9
2.4 아이디 중복 체크	10
2.5 로그인 기능	10
2.6 유저 검색	11
2.6.1 유저의 아이디를 기반으로 해당 유저의 기본 정보 가져오기	11
2.6.2 유저의 전화번호를 기반으로 해당 유저의 기본 정보 가져오기	12
3. 시스템 정보 모듈	13
3.1 시스템 정보 모듈의 사용절차	13
3.1.1 시스템 정보 등록	13
3.1.2 게임 내에서의 사용	13
3.2 버전 등록 하기	13
3.3 버전 체크하기	14

3.4 기본 건물 정보 등록하기	15
3.5 건물 정보 업데이트 하기	15
3.6 밭 정보 등록하기	17
3.7 밭 정보 업데이트 하기	17
4. 유저 마을 정보 관련	19
4.1 개요	19
4.2 유저의 마을 등록	19
4.3 마을의 자원 획득 적용하기	19
4.4 마을 정보 얻기	20
5. 친구 관리	21
5.1 친구 맺기	21
5.2 친구 마을 정보 가져오기	21
5.3 친구 삭제하기	22
6. 마을 건물관련 API	24
6.0 소개	24
6.1 건물 짓기 API	25
6.2 건물의 위치 및 방향 변경	26
6.3 유저 마을의 건물 제거하기	26
6.4 유저 마을의 건물 정보 얻어오기	27
6.5 마을 전체의 상태 업데이트 및 정보 가져오기	28
6.6 한 건물의 상태 업데이트 및 건물 정보 가져오기	28
7. 건물의 개인 생산 과정	30
7.0 서문	30

7.1 일반적인 건물의 생산 완료 요청.....	31
7.2 밭에 작물 심기.....	31
7.3 현재 생산 중인 작물 리스트 가져오기	32
7.4 작물 수확하기.....	32
8. 친구 건물의 생산 의뢰하기	33
8.0 건물 생산 의뢰의 순서	33
8.1 의뢰 가능한 상태 확인	34
8.2 건물 의뢰 신청.....	34
8.3 건물 주(소유 주) 확인.....	34
8.4 건물 주 생산 확인.....	35
8.5 의뢰 전체 완료.....	36
8.6 의뢰 목록 확인하기	36
9. 건물의 State 업그레이드 및 State 확인하기	38
9.0 서문.....	38
9.1 건물 건설이 완료되었는지 확인하기.....	39
9.2 개인 생산이 완료되었는지 확인하기.....	39
9.3 친구가 요청한 생산이 완료되었는지 확인하기.....	39
9.4 작물의 생존 시간이 지났는지 확인하기.....	40
9.5 API 사용에 대한 고찰.....	40
부록서 A: 전체적인 SNG 흐름	41
A.1 로그인 및 회원가입 흐름	41
A.2 서버 정보 업데이트 흐름	41
A.3 건물 건설의 흐름	42

A.4 건물(밭 제외)의 개인 생산 흐름	42
A.5 밭의 개인 생산 흐름	42
부록서 B: 서버 시간 얻어오기	43
부록서 C: 친구조문 대비문 표	44

1. 시작하기에 앞서

1.1 이 문서에 대하여

이 문서는 소프트웨어 마에스트로 3기 1차 프로젝트 SNG 프로토 타입 개발에서 사용하였던 SNG 서버를 제작하면서 서버에서 제공하였던 기능들의 사용 방법과 간단한 예시를 제공하기 위하여 제작한 문서입니다. 이 서버를 제작한 소스는 기본적으로 **MIT License**에 따릅니다.

MIT License의 기준은 다음과 같습니다.

1. 이 소프트웨어를 누구라도 무상으로 무제한하게 취급해도 좋다. 단, 저작권 표시 및 이 허가 표시를 소프트웨어의 모든 복제물 또는 중요한 부분에 기재해야 한다.
2. 저자 또는 저작권자는 소프트웨어에 관해서 아무런 책임을 지지 않는다.

1.2 개발 환경

이 SNG 서버 프로토타입은 다음과 같은 환경에서 제작되었습니다.

- PC: 2012년 맥북 에어
- OS: Mac OSX 10.8.1
- 개발 툴: 이클립스
- 서버 환경: 구글 앱 엔진 1.7.1

1.3 API 테스트 페이지

SNG – Prototype API에 대한 입출력을 실시간으로 확인해볼 수 있는 홈페이지를 제공합니다. 해당 폼을 통해 기본적인 입출력을 시험해보실 수 있습니다.

주소: <http://swmaestros-sng.appspot.com/>

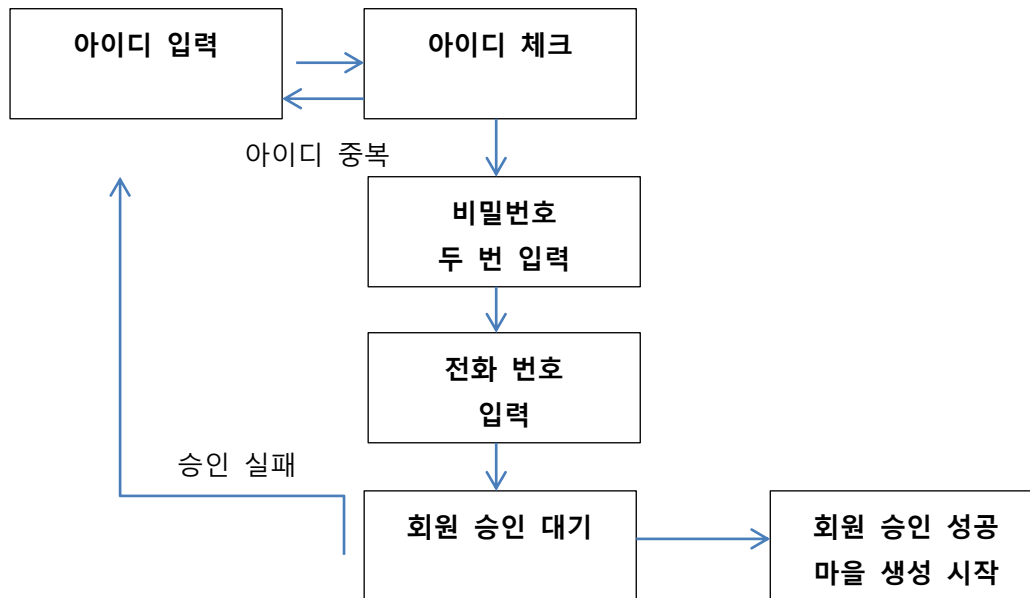
위의 홈페이지를 접속하시면 기본적인 테스트 모듈을 통해 현재 서버 진행상황과 개발을 위한 기본 감을 잡으실 수 있습니다.

1.4 관련 자료 다운로드

이 프로젝트에 대한 모든 자료는 <https://github.com/swmaestro/MobileSNG>를 통해 다운 받으실 수 있습니다.

2. 회원가입 모듈

2.1 게임에서의 회원 가입 진행 방식



2.2 기본 회원 가입 UI

이 게임에서 요구한 기본적인 회원 가입 UI는 다음과 같다. 여기에서 패스워드 확인 부분은 경우에 따라서 생략해도 문제 없다.

The UI form is a blue rectangular box containing several input fields and buttons. On the left side, there are four labels: '아이디' (ID), '패스워드' (Password), '확인' (Confirm), and '핸드폰' (Phone). To the right of these labels are four corresponding input fields. At the bottom of the form, there are two buttons: '가입하기' (Sign up) and '취소하기' (Cancel).

2.3 회원 가입

2.3.1 이전 회원 가입 스타일

SNG 프로토 타입에 가입하기 위해서는 크게 아이디, 비밀번호, 휴대전화 번호를 등록해야 합니다. 이 때 아이디는 사용자끼리 같은 값을 가져서는 안됩니다. 또한 아이디는 최대 10글자이내로 제한하여 클라이언트 표시에도 무리가 없도록 구성합니다.

변수명	타입	설명
id	String	회원의 아이디
Password	String	회원의 비밀번호
phone	String	회원의 전화번호

예제:

아이디 karuring, 비밀번호: king, phone: 010-3041-1521을 등록하는 경우

<http://swmaestros-sng.appspot.com/sngtestmodule?id=karuring&password=king&phone=010-3041-1521>

결과 XML 반환 예제

위의 결과로 실패하게되면 false, 성공하게 되면 true가 리턴된다

```
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN" "http://www.w3.org/TR/html4/loose.dtd">
<swmaestro>
  <result>false</result>
</swmaestro>
```

2.3.2 변경된 회원 가입 스타일

기존 회원 가입 스타일에서는 회원 정보를 생성한 뒤 유저 마을 정보를 따로 생성했어야 했다. 이 방식은 프로그래머의 편의에 있어서도, 그리고 클라우드 컴퓨터 입장에서 좋은 부분이 하나도 없다. 그래서 변경된 회원 가입 스타일은 두 개의 작업이 한번에 진행되도록 변경하였다.

변수명	타입	설명
id	String	회원의 아이디
Password	String	회원의 비밀번호
phone	String	회원의 전화번호
costA	Integer	마을의 초기 자원A 양
costB	Integer	마을의 초기 자원B 양

API:

<http://swmaestros-sng.appspot.com/memberregister?id=아이디>

&password=비밀번호 &phone=전화번호 &costA=초기 A자원&costB= 초기 B 자원

예제:

<http://swmaestros-sng.appspot.com/memberregister?id=karu&password=1234qwer&phone=010-3041-1521&costA=100&costB=100>

```
<swmaestro>  
  <result>true</result>  
</swmaestro>
```

2.4 아이디 중복 체크

회원가입 진행 시 아이디가 중복되어있으면 제대로 회원 가입이 불가능하다. 그래서 이 API에서는 아이디가 중복되어있는지 체크할 수 있는 기능을 제공합니다. 사용법은 다음과 같습니다.

API: <http://swmaestros-sng.appspot.com/signupnametest?id=>체크할 아이디명

예제: karu라는 아이디가 있는지 없는지 체크하는 방법

<http://swmaestros-sng.appspot.com/signupnametest?id=karu>

결과:

```
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN" "http://www.w3.org/TR/html4/loose.dtd">
<swmaestro>
    <result>false</result>
</swmaestro>
```

2.5 로그인 기능

해당 유저가 게임에 로그인하기 위해서는 아이디와 패스워드를 입력하는 로그인 과정을 거쳐 아이디와 비밀번호가 일치하면 해당 유저로 로그인 되고, 아니면 로그인을 실패하게 된다. 로그인을 하게 되면 자동적으로 플레이어의 최근 접속 일이 갱신된다.

API: <http://swmaestros-sng.appspot.com/memberlogin?id=>아이디 &password=비밀번호

예제: karu라는 아이디와 karuring이라는 패스워드로 로그인하기

<http://swmaestros-sng.appspot.com/memberlogin?id=karu&password=karuring>

결과: 가장 기본적인 result 형태로 결과가 온다. 아이디와 패스워드가 일치하면 true아니면 false

```
▼<swmaestro>
    <result>true</result>
</swmaestro>
```

2.6 유저 검색

2.6.1 유저의 아이디를 기반으로 해당 유저의 기본 정보 가져오기

해당 유저에 대한 기본정보, 전화번호, 비밀번호, 아이디, 가입일자에 대한 정보를 가져올 때 사용하는 옵션이다.

API: <http://swmaestros-sng.appspot.com/searchmember?id=검색할 아이디>

결과 리턴:

<count>: 검색된 유저 수(항상 1이다)

<userinfo>: 유저에 대한 정보를 제공

<id>: 유저 명

<phone>: 전화번호

<date>: 최근 접속 일

예제: karuring이라는 아이디를 검색하여 유저에 대한 정보를 가져오는 예제

<http://swmaestros-sng.appspot.com/searchmember?id=karuring>

```
▼ <swmaestro>
  <count>1</count>
  ▼ <userinfo>
    <id>karuring</id>
    <phone>010-3041-1521</phone>
    <date>2012-08-13-13-41-15</date>
  </userinfo>
</swmaestro>
```

2.6.2 유저의 전화번호를 기반으로 해당 유저의 기본 정보 가져오기

유저를 친구로 등록하기 위해서는 기본적으로 유저의 아이디를 알아야 하는데, 이는 실제 친구를 만나 물어보지 않는다면 거의 불가능한 일이다. 그래서 이 시스템에서는 유저의 전화번호를 바탕으로 사용자를 검색할 수 있는 기능을 제공하여 전화번호로 사용자의 정보를 얻어올 수 있다.

API: <http://swmaestros-sng.appspot.com/searchnumber?phone=>검색하고자 하는 전화번호

예제: 010 - 3041- 1521 이라는 번호를 가지는 사용자 검색

<http://swmaestros-sng.appspot.com/searchnumber?phone=010-3041-1521>

결과화면

```
▼<swmaestro>
  <count>1</count>
  ▼<userinfo>
    <id>karu</id>
    <phone>010-3041-1521</phone>
    <date>2012-08-20-21-49-48</date>
  </userinfo>
</swmaestro>
```

3. 시스템 정보 모듈

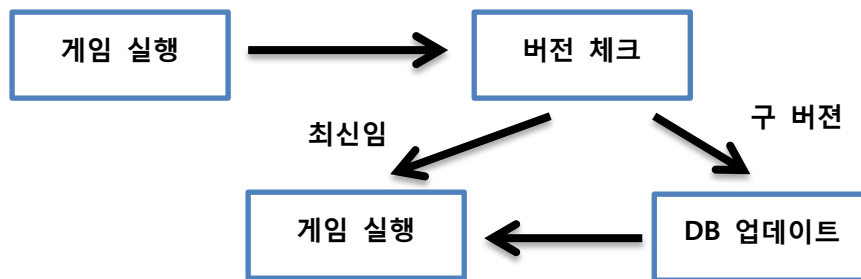
시스템 정보는 게임 내부에서 리소스들의 업데이트를 확인하기 위해 사용하는 기능이다. 시스템 정보를 게임 내에서 사용할 때는 시스템의 버전을 확인하여 내부 DB를 업데이트 하기 위한 용도로 사용된다.

3.1 시스템 정보 모듈의 사용절차

3.1.1 시스템 정보 등록

새로운 시스템 정보는
웹을 통해 업데이트

3.1.2 게임 내에서의 사용



3.2 버전 등록 하기

시스템의 버전 등록은 위에서 언급한 홈페이지에 접근하여 버전 등록 메뉴에 들어가서 정보를 입력하면 시스템의 버전이 등록된다.

버전 :

로그 :

by SNG Test

명칭	설명
버전	시스템 버전을 입력한다. 버전은 기본적으로 XXYY(X 정수부, Y 실수부)로 입력한다.
로그	해당 시스템에서 어떠한 내용을 편집했는지 입력하는 부분이다.

3.3 버전 체크하기

현재 시스템이 최신 버전인지 체크하기 위해서는 다음과 같은 API를 위해 접근하면 된다.

API: <http://swmaestros-sng.appspot.com/systemversionchk?ver=> 버전번호

결과 리턴: result 타입의 문서로 최신 버전이면 true를 아니면 false를 리턴한다.

예제: 버전 1.00이 최신인지 아닌지 테스트해보는 예제

API 주소: <http://swmaestros-sng.appspot.com/systemversionchk?ver=100>

리턴 데이터

```
<?xml version="1.0"?>
```

```
<!DOCTYPE html SYSTEM "http://www.w3.org/TR/html4/loose.dtd" PUBLIC "-//W3C//DTD HTML  
4.01 Transitional//EN">
```

```
<swmaestro>
```

```
  <result>true</result>
```

```
</swmaestro>
```

3.4 기본 건물 정보 등록하기

기본적인 건물 정보는 시스템 정보와 같이 기본적으로는 웹을 통하여 데이터를 입력하고 관리한다.

SNG 빌딩 정보 입력 기 by SNG Test

버전 :

건물고유 번호 :

건물명 :

생성금 A :

생성금 B :

수확금 A :

수확금 B :

생성시 경험치 :

생산시 경험치 :

생성 시간 :

생산활동 시간 :

크기 :

제한레벨 :

명칭	설명
버전	해당건물이 업데이트된 버전을 입력
건물 번호	건물의 기본 번호를 입력
건물 명	건물의 명칭을 입력
생성금 A	해당 건물을 생성하기 위해 자원A가 필요한 양
생성금 B	해당 건물을 생성하기 위해 자원B가 필요한 양
수확금 A	해당 건물이 자원A를 생성하는 양
수확금 B	해당 건물이 자원B를 생성하는 양
생성 시간	건물을 짓기 위한 시간(HHMMSS형태로 입력)
활동 시간	자원을 생성하기 위해 필요한 시간 (HHMMSS형태로 입력)
크기	해당 건물의 크기를 저장한다. 크기는 HHHWWW형식으로 저장
제한 레벨	해당 건물을 짓기 위한 필요 레벨이다.
생성시 경험치	건물이 생성되었을 경우 얻는 경험치 양
생산시 경험치	건물이 생산을 했을 경우 얻는 경험치 양

3.5 건물 정보 업데이트 하기

건물 정보를 업데이트 할 때에는 기본적으로 시스템의 현재 시스템의 버전을 넘겨주게 되면 업데이트 해야 할 건물 정보들이 XML 형식으로 리턴 된다

API: <http://swmaestros-sng.appspot.com/buindingostream?ver=버전번호>

결과 리턴:

```

<count>: 건물 수
<building>: 빌딩에 대한 항목
    <index>: 건물 인덱스
    <name>: 건물 명
    <costA>: 건물을 짓기 위한 자원 A 소비량
    <costB>: 건물을 짓기 위한 자원 B 소비량
    <profitA>: 건물이 생성하는 자원 A 생산량
  
```


<16profit>: 건물이 생성하는 자원 B 생산량
<Cexp>: 건물이 생성되었을 때 얻는 경험치
<Pexp>: 건물이 생산하였을 때 얻는 경험치
<b_time>: 건설 시간
<p_time>: 자원 생산 시간
<size>: 건물 크기
<level>: 건물의 요구 레벨

예제: version이 10이상인 데이터를 가져오는 예제

<http://swmaestros-sng.appspot.com/buindingostream?ver=1>

```
▼ <swmaestro>
  <count>1</count>
  ▼ <building>
    <index>100</index>
    <name>빅장</name>
    <costA>100</costA>
    <costB>0</costB>
    <profitA>10</profitA>
    <profitB>0</profitB>
    <CExp>5.0</CExp>
    <PExp>1.0</PExp>
    <b_time>1000</b_time>
    <p_time>2000</p_time>
    <size>101101</size>
    <level>1</level>
  </building>
</swmaestro>
```

3.6 발 정보 등록하기

기본적인 발 정보는 시스템 정보와 같이 기본적으로는 웹을 통하여 데이터를 입력하고 관리한다.

SNG 농작물 정보 입력기 by SNG Test

버전 :

농작물 번호 :

농작물 명 :

생성금 A :

생성금 B :

수확금 A :

수확금 B :

생산시 경험치 :

생성 시간 :

생존 시간 :

제한레벨 :

명칭	설명
버전	해당 농작물이 업데이트된 버전을 입력
농작물번호	농작물의 기본 번호를 입력
농작물 명	농작물의 명칭을 입력
생성금 A	해당 농작물을 생성하기 위해 자원A가 필요한 양
생성금 B	해당 농작물을 생성하기 위해 자원B가 필요한 양
수확금 A	해당 농작물이 자원A를 생성하는 양
수확금 B	해당 농작물이 자원B를 생성하는 양
생성 시간	농작물이 완성까지 걸리는 시간 (HHMMSS형태로 등록)
생존 시간	농작물이 생존해있는 시간 (HHMMSS형태로 등록)
제한 레벨	해당 농작물을 심기 위한 필요 레벨이다.
생산시 경험치	건물이 생산을 했을 경우 얻는 경험치 양

3.7 발 정보 업데이트 하기

발 정보를 업데이트 할 때에는 기본적으로 시스템의 현재 시스템의 버전을 넘겨주게 되면 업데이트 해야 할 건물 정보들이 XML 형식으로 리턴 된다

API: <http://swmaestros-sng.appspot.com/cropstream?ver=> 버전명

결과 리턴:

```

<count>: 작물 수
<building>: 작물에 대한 항목
  <index>: 작물 인덱스
  <name>: 작물 명
  <costA>: 작물을 짓기 위한 자원 A 소비량
  <costB>: 작물을 짓기 위한 자원 B 소비량
  <profitA>: 작물이 생성하는 자원 A 생산량
  <17rofit>: 작물이 생성하는 자원 B 생산량
  
```

<Pexp>: 작물이 생산되었을 때 얻는 경험치
<b_time>: 작물 생성 시간
<s_time>: 작물 생존 시간
<level>: 건물의 요구 레벨

예제: version이 1보다 큰 데이터 업데이트하기

<http://swmaestros-sng.appspot.com/cropostream?ver=1>

```
▼ <swmaestro>
  <count>1</count>
  ▼ <Crop>
    <index>1</index>
    <name>사탕수수</name>
    <costA>100</costA>
    <costB>0</costB>
    <profitA>20</profitA>
    <profitB>1</profitB>
    <PExp>10.0</PExp>
    <b_time>1000</b_time>
    <s_time>5</s_time>
    <level>1</level>
  </Crop>
</swmaestro>
```

4. 유저 마을 정보 관련

4.1 개요

모든 유저들은 각자 1개의 마을을 소유하고 있다. 이 마을에는 각각의 유저마다 Level이나 자원과 같은 정보들을 기록해야 할 필요가 있다. 여기에서는 유저의 마을 정보를 생성하는 것과 경험치와 자원을 획득을 얻는 방법에 대하여 설명하겠다.

4.2 유저의 마을 등록

마을 등록은 등록할 유저의 아이디와 초기 자본량만 입력하면 나머지는 자동으로 초기화 된다. 유저의 마을 등록은 **유저가 아이디를 생성할 때**, 한번 호출해주는 것으로 마을 생성이 완료된다.

API: <http://swmaestros-sng.appspot.com/villageregister?id=유저명&costA=초기자본A&costB=초기자본B>

결과 반환: 가장 자주 사용되는 XML인 result 스타일로 반환된다.

사용시 주의 점: 현재 마을을 등록할 때 입력한 아이디에 마을이 등록되어 있어도 그 여부와 상관없이 마을이 생성된다. 즉 마을이 2개 이상 만들어질 수 있으니 사용시 주의를 기울여 주시기 바랍니다.

중요: 새로운 회원 가입에서 자체적으로 마을을 생성해줍니다. 이 기능은 최대한 쓰지 않기를 권장합니다.

예제: karu라는 아이디의 마을을 생성하기

<http://swmaestros-sng.appspot.com/villageregister?id=karu&costA=100&costB=200>

결과 생략

4.3 마을의 자원 획득 적용하기

마을에서 경험치(실수량)와 자원(정수)을 증가(또는 감소) 시키는 기능으로, 경험치를 증가시키는 경우 누적량이 100이되면 마을의 레벨이 자동으로 1 증가한다. 여기서 자원과 경험치는 양수와 음수 모두 사용 가능하다.

API : <http://swmaestros-sng.appspot.com/villageadder?id=아이디 &costA=자원 A &costB= 자원 B 정수 &exp=경험치 실수량>

결과 반환: 작업의 성공 실패를 나타내는 result 가 나온다.

예제: karu라는 아이디의 마을 자원 및 경험치 획득

<http://swmaestros-sng.appspot.com/villageadder?id=karu&costA=500&costB=20&exp=50.0>

스크린 샷 생략

4.4 마을 정보 얻기

특정 유저의 마을을 받아오는 방법으로, 아이디를 전달하면 해당 유저의 마을에 대한 기초 정보가 리턴된다.

API: <http://swmaestros-sng.appspot.com/villageinfo?id=유저명>

결과 반환:

<count>: 마을 수(기본 1개)
<village>: 마을 정보에 대한 항목
 <name>: 유저 명(마을 명)
 < costA >: 소유한 자원 A
 < costB >: 소유한 자원 B
 <level>: 현재 유저의 레벨
 <exp>: 현재 경험치

예제: karu라는 유저의 마을 정보 얻어오기

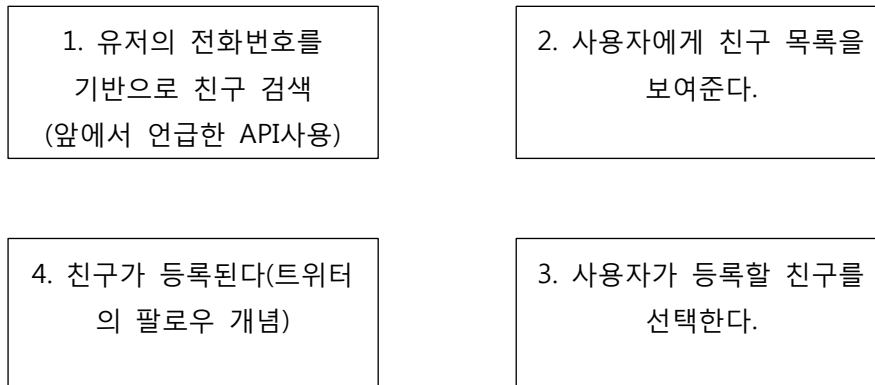
<http://swmaestros-sng.appspot.com/villageinfo?id=karu>

```
▼ <swmaestro>
  <count>1</count>
  ▼ <village>
    <name>karu</name>
    <costA>700</costA>
    <costB>240</costB>
    <level>2</level>
    <exp>0.0</exp>
  </village>
</swmaestro>
```

5. 친구 관리

5.1 친구 맺기

사용자간의 친구를 맺기 위해서는 두 유저의 아이디를 알아야 가능하다(플레이어, 친구 추가할 아이디). 그러므로 게임 내에서 친구 등록 시스템은 다음과 같은 형태로 하는 것을 추천한다.



API: <http://swmaestros-sng.appspot.com/friendadder?id=유저 아이디&friend=친구 아이디>

결과 반환: 성공과 실패 여부를 알려주는 result 타입을 이용한다.

예제: karu라는 유저가 kk88을 친구로 등록

<http://swmaestros-sng.appspot.com/friendadder?id=karu&friend=kk88>

5.2 친구 마을 정보 가져오기

현재 사용자가 친구 등록한 유저들의 마을 정보를 뽑아오는 기능이다. 이를 바탕으로 다른 유저들의 마을 정보를 살펴볼 수 있다. 이 때 Type의 값을 1을 주게 되면 유저가 등록한 친구 목록을 가져오고, 2를 넘겨주면 유저를 등록한 친구를 넘겨주게 된다.

그런데 친구가 많은 사람들은 한 화면 내에서 해당 친구 목록들을 한꺼번에 표기하기에는 메모리도 그렇고, 처음 데이터를 로딩하는데 많은 시간이 걸린다. 그래서 여기에서는 한번에 10명의 친구 정보만을 가져오고, 나중에 다른 사람들을 10명 단위로 가져올 수 있다.

API: [http://swmaestros-sng.appspot.com/friendinfo?id=유저명&type=타입 넘버&page=페이지\(1부터 시작\)](http://swmaestros-sng.appspot.com/friendinfo?id=유저명&type=타입 넘버&page=페이지(1부터 시작))

결과 반환:

<count>: 마을 수

<village>: 마을 정보에 대한 항목

<name>: 유저 명(마을 명)
< costA >: 소유한 자원 A
< costB >: 소유한 자원 B
<level>: 현재 유저의 레벨
<exp>: 현재 경험치

예제1: karu라는 유저가 추가한 친구 정보

<http://swmaestros-sng.appspot.com/friendinfo?id=karu&type=1&page=1>

```
▼<swmaestro>
  <count>1</count>
  ▼<village>
    <name>kk88</name>
    <costA>100</costA>
    <costB>200</costB>
    <level>1</level>
    <exp>0.0</exp>
  </village>
</swmaestro>
```

예제2: karu라는 유저를 등록한 친구 정보

<http://swmaestros-sng.appspot.com/friendinfo?id=karu&type=2&page=1>

//kk88은 karu라는 유저를 등록하지 않은 상태

```
▼<swmaestro>
  <count>0</count>
</swmaestro>
```

5.3 친구 삭제하기

친구간의 팔로잉 관계를 끊고 싶을 때는 친구 삭제하기 기능을 이용하여 친구간의 관계를 지울 수 있다.

API: <http://swmaestros-sng.appspot.com/frienddelete?id= 유저 아이디 &friend= 지울 친구 아이디>

결과 반환: 작업의 성공 여부를 알려주는 result타입으로 리턴된다.

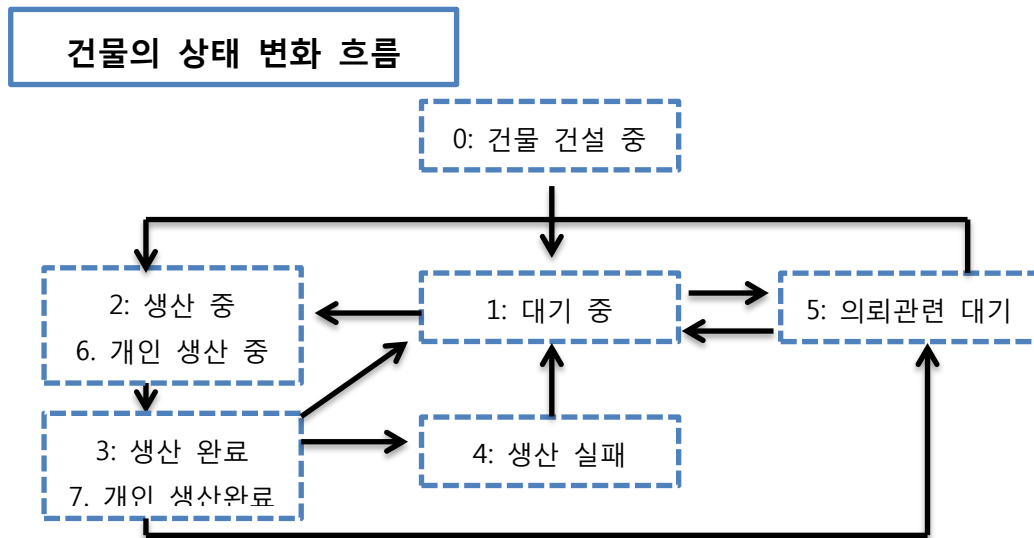
예제: <http://swmaestros-sng.appspot.com/frienddelete?id=Sunas&friend=test3>

```
▼<swmaestro>  
  <result>true</result>  
</swmaestro>
```


6. 마을 건물관련 API

6.0 소개

마을 건물 관련 API에서는 유저의 마을에 건물을 짓거나, 이미 지어진 건물들의 상태를 변경할 수 있는 기능, 그리고 해당 유저의 마을에 빌딩 정보들을 가져올 수 있는 기능들을 제공해준다. 그런데 이러한 SNG에서의 건물들은 시간의 흐름에 따라서 건물의 상태(발도 유사하게 진행된다)가 변하게 된다. 이 게임에서도 총 6개의 상태를 가지고 있으며, 시간이나 다양한 흐름에 따라서 상태가 변경된다.



상태	설명
0: 건물 건설 중	현재 건물을 건설하고 있는 중이다. 이 단계는 정해진 생성 시간이 지나게 되면 건물에게 경우는 자동으로 개인 생산 중으로, 발의 경우 대기 중으로 변경된다.
1: 대기 중	현재 발이 다른 작업을 위해 대기하고 있는 상태이다.
2: 의뢰 생산 중	현재 건물이 친구에게 의뢰 받은 물품을 생성하고 있는 단계이다. 일정시간이 지나면 생산 완료 단계로 넘어간다.
3: 의뢰 생산 완료	의뢰 생산이 모두 완료되어 유저가 확인하기를 대기하는 상태이다.
4: 생산 실패	생산 완료 후 수확 시간이 지나 생산이 실패가 된 상태이다. 이 상태는 발만 갈 수 있다.
5: 의뢰 관련 대기	건물(발 제외)이 다음 작업을 위해 대기하고 있는 상태이다.
6: 개인 생산 중	건물이나 발이 개인적인 생산(다른 유저에게 부탁 받은 것 제외)을 진행하고 있는 상태이다. 이 상태일 때 친구의 의뢰를 받을 수 있으며 의뢰를 받게 되면 개인 생산이 중단된다.
7: 개인 생산 완료	건물이나 발이 개인적인 생산을 완료한 상태이다. 그리고 발은 이 상태에서 정해진 생존 시간이 지나게 되면 생산 실패 상태로 변경된다.

그러나 우리의 게임 서버는 Http 기반으로 이루어지는 서버이기 때문에 지속적으로 서버가 시간을 감시하면서 내용을 변경시킬 수 없다. 그렇기 때문에 이 API에서는 건물의 상태의 갱신하는 방법으로 유저 마을 전체를 업그레이드 하는 방법과 건물 단위로 상태를 업그레이드 하는 방법, 총 두 가지를 제공한다(현재 전체 업그레이드만을 지원 곧 건물 단위 지원 예정).

6.1 건물 짓기 API

일반적인 SNG에서 유저는 각자 자신에게 부여된 마을에 다양한 건물(밭 포함)들을 건설할 수 있다. 그렇기 때문에 이 게임에서도 유저의 마을에 건물을 지을 수 있는 API를 제공한다. 그 API는 다음과 같다.

API: <http://swmaestros-sng.appspot.com/buildinginsert?id=유저명&bindex=건물타입번호&index=건물 고유 번호 &location=위치값&direction=방향성>

값(타입)	설명
id(String)	건설할 마을의 유저명을 넘겨준다.
bindex(int)	건설할 건물 종류 값을 넘겨준다.
index(int)	건물의 고유 값을 지정해준다(일반 건물 0~999, 밭 1000~).
location(int)	건물의 위치값을 지정해준다.
Direction(Boolean)	건물의 방향성을 true, false로 지정해준다.

결과 반환: 가장 기본적인 result 타입으로 성공여부를 알려준다.

주의:

참고로 건물을 건설하게 되면 자동으로 건물 종류의 지정된 값에 따라서 건물의 상태가 건설 중으로 변경되며, 자동으로 필요한 자원 소비량이 차감된다. 이 때 자원의 양이 -로 넘어가는지는 서버에서 체크하지 않는다. 추가로 1.0 버전부터는 건물 생성 시 생성 경험치도 자동으로 증가한다(룰 더 스카이 시스템 차용). 또한 밭을 제외한 건물들은 생성이 완료되면 대기 상태가 아닌 개인 자원 생산 상태로 돌입하여 자동적으로 자원을 생산한다. 또한 이 과정 중 친구의 의뢰를 받으면 생산이 중지된다.

예제:

<http://swmaestros-sng.appspot.com/buildinginsert?id=karu&bindex=100&index=1&location=1001&direction=true>

```
▼<swmaestro>
  <result>true</result>
</swmaestro>
```

6.2 건물의 위치 및 방향 변경

마을의 건물을 건설한 다음, 차후의 건물의 위치를 바꾸거나 건물의 방향을 바꿀 필요가 있을 수 있다. 그러므로 여기에서는 이미 건설된 건물의 위치 및 방향을 바꾸는 기능을 제공한다.

API: [http://swmaestros-sng.appspot.com/buildinglchange?id=아이디&index=건물의고유번호
&location=건물의 변경될 위치 &direction= 건물의 방향](http://swmaestros-sng.appspot.com/buildinglchange?id=아이디&index=건물의고유번호&location=건물의 변경될 위치 &direction= 건물의 방향)

결과 반환: 실행이 정상적으로 되었는지 확인할 수 있도록 가장 기본적인 <result> 형태를 이용한다.

예제:

<http://swmaestros-sng.appspot.com/buildinglchange?id=karu&index=1&location=1002&direction=false>

```
▼<swmaestro>  
  <result>true</result>  
</swmaestro>
```

6.3 유저 마을의 건물 제거하기

유저 마을의 건물을 건설한 다음, 유저는 자신의 의지로 건물을 제거할 수 있다. 하지만 이 때 건물을 제거하게 되면 건물을 건설했을 때 소모한 자원은 다시 돌려 받지 못한다.

API: <http://swmaestros-sng.appspot.com/vbdelete?id=유저 아이디&index=건물 고유 번호>

결과 반환: 성공 실패 여부를 알려주는 result 타입이 반환된다.

예제: karu라는 유저의 고유 번호 1번을 삭제하는 예제

<http://swmaestros-sng.appspot.com/vbdelete?id=karu&index=1>

```
▼<swmaestro>  
  <result>true</result>  
</swmaestro>
```

6.4 유저 마을의 건물 정보 얻어오기

SNG에서는 자신의 마을은 물론 친구들의 마을에도 방문을 해야 하기 때문에 유저 마을의 건물 정보를 얻어올 수 있는 기능이 제공되어야 한다. 그러므로 이 API에서도 유저 마을에 건설된 건물들에 대한 정보를 주는 기능을 제공한다.

API: <http://swmaestros-sng.appspot.com/vbinfolist?id=아이디>

결과 반환:

<count>: 마을 건물들의 정보의 개수

<building>:

-<index>: 건물의 고유 번호

-<bindex>: 건물의 종류 번호

-<state>:

0: 현재 건물 건설중

1: 대기 중(건설이 끝나고 작업을 기다림)

2: 의뢰 생산 중

3: 의뢰 생산 완료 대기

4: 생산 실패(대기 시간 초과)

5: 의뢰관련 대기 중(받은 제외)

6: 개인 생산 중

7: 개인 생산 완료 대기

-<location>: 건물의 위치 XXXYYY형태로 입력하자

-<direction>: 건물의 방향을 나타낸다. True, false 형태의 값을 가진다.

-<date>: 가장 최근 업데이트된 날짜이다.

예제: karu라는 유저의 정보를 얻어오기

<http://swmaestros-sng.appspot.com/vbinfolist?id=karu>

```
<swmaestro>
  <count>1</count>
  <building>
    <index>1</index>
    <bindex>1</bindex>
    <state>0</state>
    <location>1002</location>
    <direction>false</direction>
    <date>2012-08-27-12-14-06</date>
  </building>
</swmaestro>
```

6.5 마을 전체의 상태 업데이트 및 정보 가져오기

게임을 종료했다가 나중에 다시 킨 경우(즉 게임을 처음 실행한 경우)나 다른 유저가 친구의 맵을 방문할 때, 게임 내 건물 및 밭들의 상태를 갱신해줄 필요가 있다. 그래서 여기에서는 데이터를 갱신해준 다음 건물 정보를 얻어올 수 있는 기능들을 제공한다.

API : <http://swmaestros-sng.appspot.com/vbstateupdate?id=아이디>

결과 반환: 6.4에서 소개한 <building> 타입을 리턴한다.

참고: 여기서 상태 업데이트는 건물의 건설 완료, 생산의 완료여부, 작물의 생존 시간 초과 등 이 게임에서 기본적인 상태들의 변경여부를 체크한 뒤 갱신한다.

예제: karu라는 유저의 정보를 갱신하고 얻어오기

<http://swmaestros-sng.appspot.com/vbstateupdate?id=karu>

```
▼<swmaestro>
  <count>1</count>
  ▼<building>
    <index>1</index>
    <bindex>1</bindex>
    <state>1</state>
    <location>1001</location>
    <direction>true</direction>
    <date>2012-08-28-19-28-40</date>
  </building>
</swmaestro>
```

6.6 한 건물의 상태 업데이트 및 건물 정보 가져오기

6.5절에서 설명한 내용은 마을 전체의 내용을 갱신하고, 그에 대한 갱신된 정보를 가져오는 방법이었다. 그러나 마을 전체를 갱신하고 그 정보를 받아오는 것은 서버와 클라이언트, 모두에게 부담되는 방법이다. 그래서 마을 전체의 업데이트 대신 특정 건물만을 업데이트 하는 기능을 제공하였다.

API: <http://swmaestros-sng.appspot.com/onestateupdate?id=유저 아이디&index=건물 고유 번호>

결과 반환: 6.4에서 소개한 <building>을 리턴한다.

주의: 현재 존재하지 않는 건물의 고유 번호를 입력하면 에러가 발생하니 주의하시길 바랍니다. 그외 실제 사용에는 다른 이상이 없습니다.

예제: karu라는 유저의 3000(발)건물의 상태를 업데이트 한다.

<http://swmaestros-sng.appspot.com/onestateupdate?id=karu&index=3000>

```
▼<swmaestro>
  <count>1</count>
  ▼<building>
    <index>3000</index>
    <bindex>-1</bindex>
    <state>1</state>
    <location>1002</location>
    <direction>false</direction>
    <date>2012-09-03-00-06-49</date>
  </building>
</swmaestro>
```

7. 건물의 개인 생산 과정

7.0 서문

SNG에서의 건물이 생산하는 방식은 크게 3개의 방식이 존재한다. 하나는 밭에 작물을 심는 방식, 두 번째는 건물이 건설 된 후에 자동으로 생산하는 방식, 그리고 마지막으로 친구의 건물에 생산을 의뢰하는 방식이 존재한다. 앞에서 이야기한 두 가지 방식은 개인이 진행하는 방식이며, 마지막 방식은 다른 사람과 협동적으로 생산하는 방식이다. 여기에서는 개인이 생산을 진행하는 과정에 대해서만 이야기하겠다.

건물의 3가지 생산 방식

생산 방식	협동 유무	설명 및 절차
건물 자동 생산	개인	<p>설명: 건물이 건설된 뒤 자동으로 자원을 생성하는 방식이다. 자동 생산을 진행하고 있을 때, 다른 친구에게 의뢰를 받게 되면 생산은 자동 중지된다. 또한 생산이 완료되면 소유자가 확인할 때까지 생산이 멈추며, 소유자가 확인하면 다시 생산이 시작된다.</p> <p>대상: 모든 건물(밭 제외)</p> <p>과정:</p> <ol style="list-style-type: none">기본 과정 건물이 건설됨 -> 생산 시작 -> 생산 완료 및 확인 ->(자동) 생산 시작 -> 생산 완료 및 확인 -> 반복친구의 의뢰가 온 경우 생산 시작 -> 친구의 의뢰 ->의뢰확인 및 처리 ->의뢰 완료 후 다시 생산 시작
작물 생산	개인	<p>설명: 밭에다가 추가적인 작물을 심어서 생산하는 방식이다. 밭의 경우 건물과 달리 예외를 가지지 않는다. 밭을 통한 작물 생산은 건물과 달리 소유자가 작물을 심어야 생산이 시작된다. 또한 특정 시간이 지나도록 수확하지 않으면 생산물의 수확을 할 수 없다.</p> <p>대상: 밭</p> <p>과정:</p> <ol style="list-style-type: none">기본 과정 밭 생성 -> 심을 작물 선택 -> 작물이 자라남 -> 작물 생산 완료 -> 소유자 확인 ->반복
의뢰 생산	협동	<p>설명: 친구에게 의뢰를 받아 생산하는 방식 이에 대해서는 8장에서 따로 설명하겠다.</p>

참고: 건물이 생산하던 중 생산시간이 다되었다면 상태를 갱신해주어야만 생산이 모두 완료되었는지 체크할 수 있다. 갱신을 하지 않는다면 생산 완료 시간이 지나도 상태가 그대로이다.

7.1 일반적인 건물의 생산 완료 요청

밭을 제외한 다른 건물들은 자동적으로 건물의 생산을 시작한다. 그러나 생산 완료의 경우에는 소유주가 확인을 해야만 모든 과정이 마무리가 된다. 그래서 여기에서는 건물이 생산을 완료한 상태일 때 생산 완료를 처리하는 API를 소개하겠다. 참고로 건물의 생산이 완료되면 자동으로 지정된 경험치와 자원이 소유자의 자금에 더해진다. 그러므로 따로 서버에 Add 기능을 호출하지 않아도 된다.

사용 조건: 특정 건물의 state가 개인 생산 완료일 때만 가능

API: http://swmaestros-sng.appspot.com/ap_requestend?id=아이디&index=건물 고유 번호

결과 반환: 결과를 알려주는 <result> 타입이 반환된다.

예제: karu라는 유저 마을의 111 번호의 건물에서 완료를 체크하는 과정

http://swmaestros-sng.appspot.com/ap_requestend?id=karu&index=111

그림 생략

7.2 밭에 작물 심기

건물과 달리 밭은 소유자가 밭의 특정 작물을 심어야만 생산이 시작된다. 그래서 밭의 경우는 밭에 작물을 심는 것에 대하여 설명하겠다.

사용 조건: 건물(밭)의 상태가 대기 중으로 있어야만 한다.

API: http://swmaestros-sng.appspot.com/crop_request?id=아이디 &index= 밭의 고유 번호 &cindex=작물 고유 번호

결과 반환: 결과를 알려주는 <result> 타입이 반환된다.

예제: karu라는 유저 마을의 3000 번호의 밭에 0번 작물을 심는 과정

http://swmaestros-sng.appspot.com/crop_request?id=karu&index=3000&cindex=0

그림 생략

7.3 현재 생산 중인 작물 리스트 가져오기

건물 리스트를 가져올 때 건물의 종류에 대해서만 가져올 뿐 밭의 경우 현재 밭에 심어진 작물 리스트를 가져오지 않는다. 그래서 건물리스트와는 다르게 현재 작물이 심어져있는 밭들의 고유 값과 작물 번호를 가져오는 기능을 추가적으로 제공한다. **그러므로 맵 로딩이 이루어질 때 추가적으로 작물 리스트도 같이 이용해야만 한다.**

API: http://swmaestros-sng.appspot.com/crop_rlist?id=아이디

결과 반환:

- <count>: 생산 중인 작물 수
- <crop>:
 - <index>: 작물을 생산하고 있는 밭의 고유 번호
 - <cindex>: 생산 중인 작물 고유 값

예제: karu라는 유저가 생산 중인 작물 리스트

http://swmaestros-sng.appspot.com/crop_rlist?id=karu

```
▼<swmaestro>
  <count>1</count>
  ▼<crop>
    <index>3000</index>
    <cindex>0</cindex>
  </crop>
</swmaestro>
```

7.4 작물 수확하기

생산 중인 작물이 생산이 완료되면 유저가 생산 완료를 확인해야 모든 생산이 마무리 된다. 그러면 자동적으로 작물에 지정된 자원과 경험치가 유저에게 반영된다. 그러나 지정된 생존 시간 안에 확인하지 못하여 생산 실패가 된 경우 그 때에도 이 API를 이용해 정리하여야 하지만 그 때에는 경험치가 반영되지 않는다.

사용 조건: 밭의 상태가 개인 생산 완료 또는 생산 실패 단계

API: http://swmaestros-sng.appspot.com/crop_rend?id=아이디 &index=밭의 고유 번호

결과 반환: 결과를 알려주는 <result> 타입이 리턴된다.

예제: karu라는 유저의 3000번째 밭 수확

http://swmaestros-sng.appspot.com/crop_rend?id=karu&index=3000

그림 생략

8. 친구 건물의 생산 의뢰하기

8.0 건물 생산 의뢰의 순서

건물은 발과는 달리 친구(팔로우 관계)의 요청을 받아, 건물이 아이템을 생산하고, 그 후 건물의 소유주와 요청한 친구의 경험치와 자금이 올라가는 다소 복잡한 과정을 걸쳐서 진행된다. 그렇기 때문에 이에 대한 API를 소개하기 전에 우선 건물 생산이 이루어지는 과정에 대해서 소개한다.

건물 생산 의뢰의 진행 절차

1. 친구가 내 마을에 방문한다.	2. 친구가 마을 내 건물을 선택하여 의뢰를 신청	3. 의뢰를 건물주가 승인한다.
4. 건물이 의뢰를 생산한다.	5. 건물이 의뢰를 생산 완료한다.	6. 건물이 의뢰를 생산 완료한다.
6. 소유자가 생산 완료를 확인한다.	7. 친구가 생산 완료를 승인 후 각자 경험치, 돈을 얻음	

API 명	사용 단계	설명
건물 의뢰 신청	2	친구가 소유자의 마을을 방문한 다음 특정 건물을 선택하여 의뢰를 신청할 수 있다. 의뢰는 친구 당 하나씩 만 가능하다(체크 로직 제공 예정). API 사용처: 친구가 상대 마을에 방문했을 때
건물 주 확인	3	친구가 의뢰를 요청하면 건물주가 해당 내용을 승인해야 한다. 건물 주는 해당 요청이 온 다음 승인 버튼을 눌러야 생산 단계로 넘어간다. API 사용처: 친구의 요청을 승인할 때
건물 주 생산 확인	6	건물이 생산을 완료하게 되면 플레이어가 생산 내용을 확인해야 다음 단계로 넘어가게 된다. 즉 플레이어가 생산이 완료됨을 승인해야 7번 단계로 넘어가게 된다. API 사용처: 건물 주가 생산 내역을 확인할 때(건물 터치!)
의뢰 전체 완료	7	요청한 친구가 의뢰가 완료된 것을 확인한 다음 요청한 친구와 건물 소유자의 자금과 경험치가 정해진 양만 큼 증가한다. API 사용처: 요청한 친구가 의뢰 확인을 할 때

주의 점: 이전 단계가 완료되지 않으면 이후 단계가 진행되지 않는다.

8.1 의뢰 가능한 상태 확인

유저간의 의뢰는 1:1로 한 개씩만 가능하다. 그렇기 때문에 여기에서는 유저가 건물의뢰를 부탁할 수 있는 상태 있는지 체크할 수 있는 기능을 제공한다.

API: <http://swmaestros-sng.appspot.com/isrequest?owner=소유자 아이디 &requester=요청자 아이디>

결과 반환: 성공 여부를 알려주는 <result> 타입이 리턴된다.

예제: karu라는 소유자에게 kk88이 요청할 수 있는지 체크하는 로직

<http://swmaestros-sng.appspot.com/isrequest?owner=karu&requester=kk88>

```
▼ <swmaestro>
  <result>true</result>
</swmaestro>
```

8.2 건물 의뢰 신청

건물에 의뢰를 신청하는 단계에서 사용하는 API는 다음과 같다.

API : http://swmaestros-sng.appspot.com/brequest_create?owner=소유자&requester=요청자아이디 &index=건물고유번호

결과 반환: 성공 여부를 알려주는 <result> 타입이 리턴된다.

예제: kk88이라는 사람이 karu의 마을 1번 건물에 요청을 부탁

http://swmaestros-sng.appspot.com/brequest_create?owner=karu&requester=kk88&index=1

```
▼ <swmaestro>
  <result>true</result>
</swmaestro>
```

8.3 건물 주(소유 주) 확인

건물에 의뢰가 들어온 다음 소유자가 확인한 것을 처리하기 위한 API는 다음과 같다. 여기서 건물 주가 확인을 하게 되면 건물은 자동적으로 생산 상태로 변경된다.

API: http://swmaestros-sng.appspot.com/brequest_accept?owner=소유자 아이디&index=건물 고유번호

결과 반환: 성공 여부를 알려주는 <result> 타입이 리턴된다.

예제: karu의 마을 1번 건물에 요청을 수락

http://swmaestros-sng.appspot.com/brequest_accept?owner=karu&index=1

```
▼<swmaestro>  
  <result>true</result>  
</swmaestro>
```

8.4 건물 주 생산 확인

건물이 생산을 완료하게 되면 건물 주가 생산 완료(이전에 상태 업그레이드를 해주어야 함: 6장에서 설명하는 건물 상태업그레이드 기능 이용)를 확인 해야 한다. 확인하기 위한 API는 다음과 같다.

API: http://swmaestros-sng.appspot.com/brequest_productend?owner=소유자&index=건물 고유번호

결과 반환: 성공 여부를 알려주는 <result> 타입이 리턴된다.

예제: karu의 마을 1번 건물에 요청된 생산 완료를 확인

http://swmaestros-sng.appspot.com/brequest_productend?owner=karu&index=1

```
▼<swmaestro>  
  <result>true</result>  
</swmaestro>
```

8.5 의뢰 전체 완료

생산이 된 후 소유자가 승인을 하면 최종적으로 의뢰자(친구)가 승인을 하면 모든 의뢰가 완료된다. 의뢰가 완료되면 정해진 자본과 경험치를 획득한다.

API: http://swmaestros-sng.appspot.com/brequest_end?owner=소유자명 &index= 고유 번호

결과 반환: 성공 여부를 알려주는 <result> 타입이 리턴 된다

예제: kk88이 karu의 건물에 의뢰가 완료된 것을 확인한다.

http://swmaestros-sng.appspot.com/brequest_end?owner=karu&index=2.

의뢰 완료 전 자원

VillageCostA	VillageCostB	VillageExp	VillageLevel	memberId
2000	1000	-2.0	3	sng
330	80	2.0	1	karu
100	200	0.0	1	kk88

완료 후 자원

VillageCostA	VillageCostB	VillageExp	VillageLevel	memberId
2000	1000	-2.0	3	sng
360	100	7.0	1	karu
130	220	5.0	1	kk88

8.6 의뢰 목록 확인하기

유저와 친구들간의 의뢰 목록은 SNG에서 서로간의 교류의 흔적이다. 그렇기 때문에 이러한 항목들은 언제든지 확인할 수 있으면 좋을 것이라고 생각된다. 그래서 이 API에서는 의뢰 목록을 확인할 수 있는 방법을 제공한다.

API:

http://swmaestros-sng.appspot.com/brequest_list?id=유저 아이디

&state=상태 값&ord=부등호 방식 &usertype=유저 형태&page=페이지 넘버

상태 값

- 0: 요청을 받아 소유주가 승인하기를 대기하는 상태
- 1: 소유주가 승인후 작물을 생산하고 있는 상태
- 2: 작물 생산이 완료되어 의뢰자가 확인을 내리기 위한 상태
- 3: 완료된 의뢰

부등호 방식:

- true: 등호

- false: <= 형태의 등호 주어진 스테이트 값보다 작은 값들을 모두 가져온다.

유저 형태:

- true: 내가 요청 받은 내용
- false: 내가 요청한 내용

예제:

<http://swmaestros->

sng.appspot.com/brequest_list?id=karu&state=3&ord=true&usertype=true&page=1

```
<swmaestro>
  <count>2</count>
  ▼<RequestInfo>
    <owner>karu</owner>
    <requester>kk88</requester>
    <state>3</state>
    <index>1</index>
    <bindex>1</bindex>
  </RequestInfo>
  ▼<RequestInfo>
    <owner>karu</owner>
    <requester>kk88</requester>
    <state>3</state>
    <index>2</index>
    <bindex>10</bindex>
  </RequestInfo>
</swmaestro>
```

9. 건물의 State 업그레이드 및 State 확인하기

9.0 서문

건물의 상태를 업그레이드 하는 것에 대한 내용은 6.5와 6.6에서 설명한 기능을 이용하여서 상태를 업그레이드를 할 수 있다. 하지만 이 기능들은 너무 범용적(상태가 변화될 수 있는 모든 경우들을 체크하여 반영한다)이며, 결과 리턴이 해당 건물에 대한 정보를 리턴 하였기 때문에 상태 변경이 있었는지 체크하기 힘든 부분이 있다. 그래서 여기에서는 개별 상태 체크 및 업데이트를 하는 기능을 제공한다.

State 업데이트 조건

상황	설명
건물 건설이 완료 되었는지 확인	설명: 건물의 건설이 완료되었는지 확인할 필요가 있다. 건물 건설은 건물마다 건설 시간이 다르므로 해당 시간이 지났는지 확인할 필요성이 있다. 또한 건물 건설이 완료되면 일반 건물은 자동적으로 개인 생산을, 발은 대기 상태로 들어가야 한다. 상태 변화(완료된 경우): <ul style="list-style-type: none">- 일반 건물: 건설 중 -> 개인 생산 중- 발: 건설 중 -> 대기 중
개인 생산이 완료 되었는지 확인	설명: 건물이 생산하고 있는 내용이 생산이 완료되었는지 확인해야 할 필요가 있다. 이 기능은 수확을 직접적으로 행하는 것이 아닌 생산이 완료되었는지 확인하고 건물의 State를 생산 중에서 생산 완료로 바꿔주기만 한다(상태가 생산 완료가 아니면 수확 API 사용이 불가능하다). 상태 변화: <ul style="list-style-type: none">- 일반: 개인 생산 중 -> 개인 생산 완료- 발: 개인 생산 중 -> 개인 생산 완료
친구가 요청한 생산이 완료되었는지 확인	설명: 위에서 소개한 개인 생산이 완료되었는지 확인하는 것과 비슷한 기능으로 여기서는 단지 친구가 요청한 생산에 대해서만 판단한다는 것이 다르다.
작물의 생존 시간이 넘었는지 확인	설명: 작물의 생존 시간이 지나갔는지 체크하는 기능이다. 생존시간이 넘어가게 되면 자동적으로 상태가 생산 실패 상태로 변하게 된다. 상태 변화: <ul style="list-style-type: none">- 발: 생산 완료 -> 생산 실패

9.1 건물 건설이 완료되었는지 확인하기

특정 건물 건설이 완료되었는지 확인하는 API를 다음과 같이 제공한다. 이 API는 클라이언트 내에서 시간을 갱신으로 자체적으로 완성여부를 체크하다가 건물이 완성되었을 때, 한 번 호출해주면 서버에도 자동적으로 건물의 상태가 완성됨으로 바뀌게 된다.

API: http://swmaestros-sng.appspot.com/build_check?id=아이디&index=건물고유번호

결과 반환: 성공 여부를 알려주는 <result> 타입이 리턴 된다(건물의 건설이 완료되면 true)

예제: karu라는 유저의 111번 건물이 건설되었는지 확인하는 예제

http://swmaestros-sng.appspot.com/build_check?id=karu&index=111

그림 생략

9.2 개인 생산이 완료되었는지 확인하기

특정 건물이 개인 생산하고 있는 물품의 개인 생산이 완료되었는지 체크하는 기능이다. 수확하기 전에 이 API를 통해 생산이 완료되었는지 확인한 뒤 수확하기 API를 통해 수확을 진행해야 한다.

API: http://swmaestros-sng.appspot.com/product_check?id=아이디&index=건물고유번호

결과 반환: 성공 여부를 알려주는 <result> 타입이 리턴 된다(건물의 개인 생산이 완료되면 true)

예제: karu라는 유저의 111번 개인 생산이 완료되었는지 확인하는 예제

http://swmaestros-sng.appspot.com/product_check?id=karu&index=111

그림 생략

9.3 친구가 요청한 생산이 완료되었는지 확인하기

친구가 요청한 의뢰에 대한 생산은 개인 생산과 다르게 취급된다. 그래서 따로 생산을 체크하는 API가 따로 제공된다. 기본적인 기능은 9.2의 개인 생산과 유사하다.

API: http://swmaestros-sng.appspot.com/friendp_check?id=아이디&index=건물고유번호

결과 반환: 성공 여부를 알려주는 <result> 타입이 리턴 된다(건물의 생산이 완료되면 true)

예제: karu라는 유저의 111번 건물의 의뢰 생산이 완료되었는지 확인하는 예제

http://swmaestros-sng.appspot.com/friendp_check?id=karu&index=111

그림 생략

9.4 작물의 생존 시간이 지났는지 확인하기

건물과 달리 작물은 생성 완료 시, 일정 시간 안에 수확해야만 자원과 경험치를 획득할 수 있다. 그래서 이 게임에서 생존 시간이 지났는지를 확인할 필요 성이 있어 이 기능을 제공하게 되었다.

API: http://swmaestros-sng.appspot.com/fail_check?id=아이디&index=발의 고유 번호

결과 반환: 성공 여부를 알려주는 <result> 타입이 리턴 된다(작물의 생존 시간이 지나면 true)

사용 예제: karu라는 유저의 3000번 값을 가진 발의 작물이 생존해 있는지 체크

http://swmaestros-sng.appspot.com/fail_check?id=karu&index=3000

이미지 생략

9.5 API 사용에 대한 고찰

API 사용에 어려움이 있는 것 같아 State 업데이트에 대한 예를 간단히 들어 드리겠습니다.

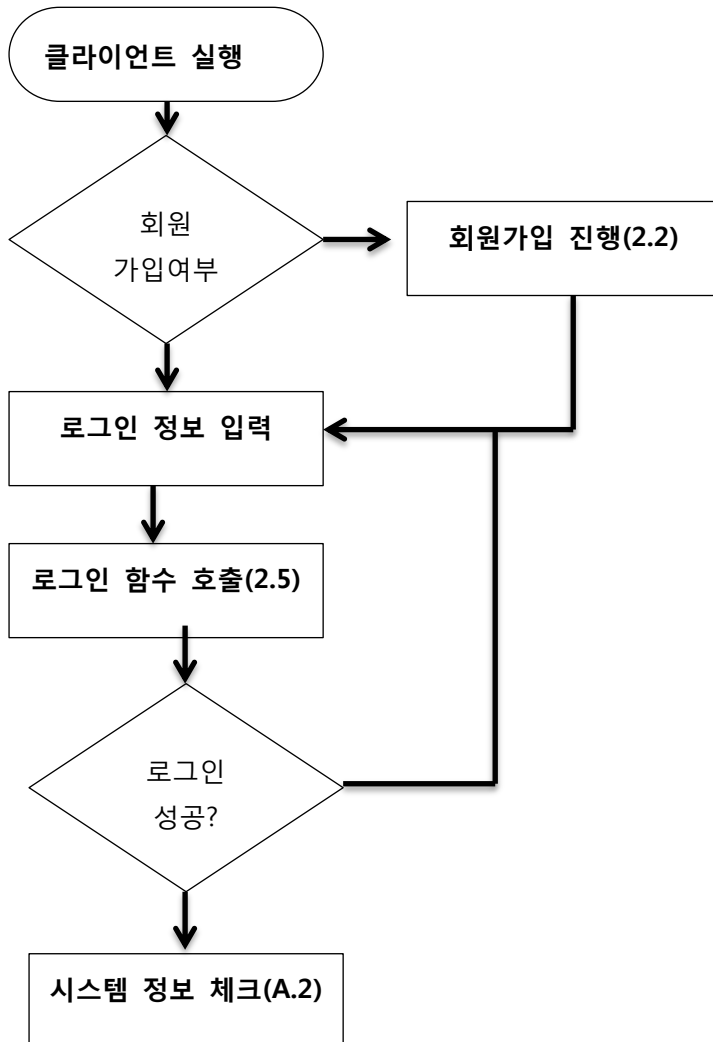
또한 9장에서 소개된 모든 기능들은 6.5와 6.6에서 소개된 API로도 대체 가능합니다. 단지 이 기능들은 성공유무와 개별적이라는 점이 다릅니다(6.5와 6.6은 내용을 모두 체크하게 됨)

발의 작물 수확 과정

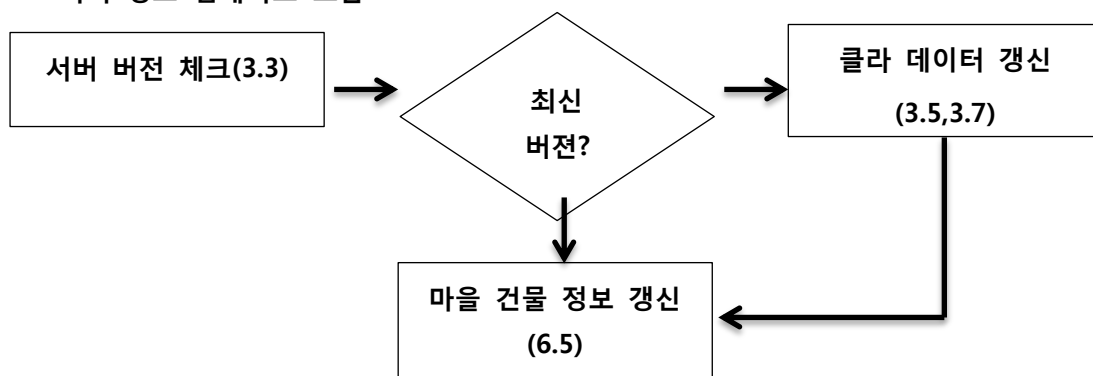
발을 생성 -> 일정 시간 후 9.1의 API로 상태 갱신 또는 6.5 및 6.6 API로 상태 갱신 -> 7.2 API를 이용해 발을 작물을 심기 -> 일정 시간 이후 발의 작물의 상태를 9.2를 통해 체크 -> true면 수확 진행, false면 대기 -> 9.4 작물의 생존 시간을 확인하고 생존 했는지 체크 -> 7.4를 이용해 수확

부록서 A: 전체적인 SNG 흐름

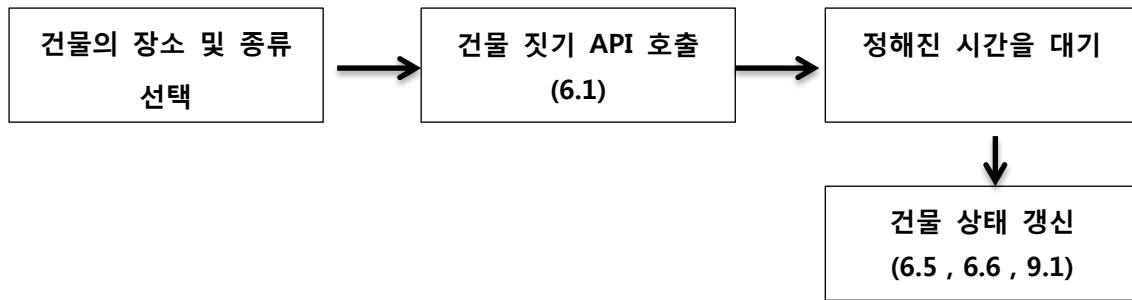
A.1 로그인 및 회원가입 흐름



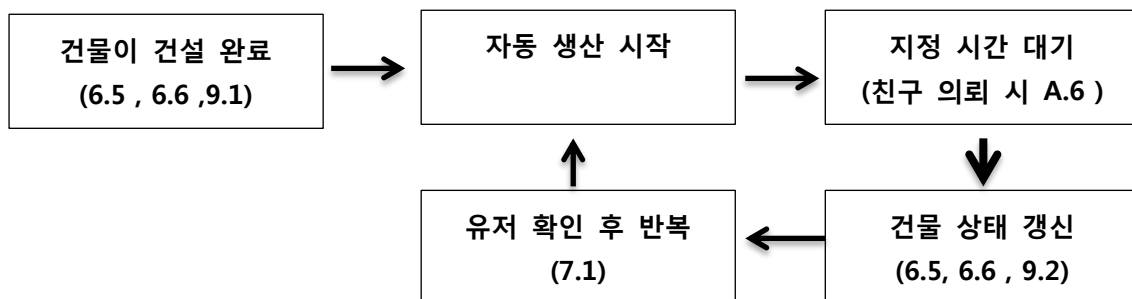
A.2 서버 정보 업데이트 흐름



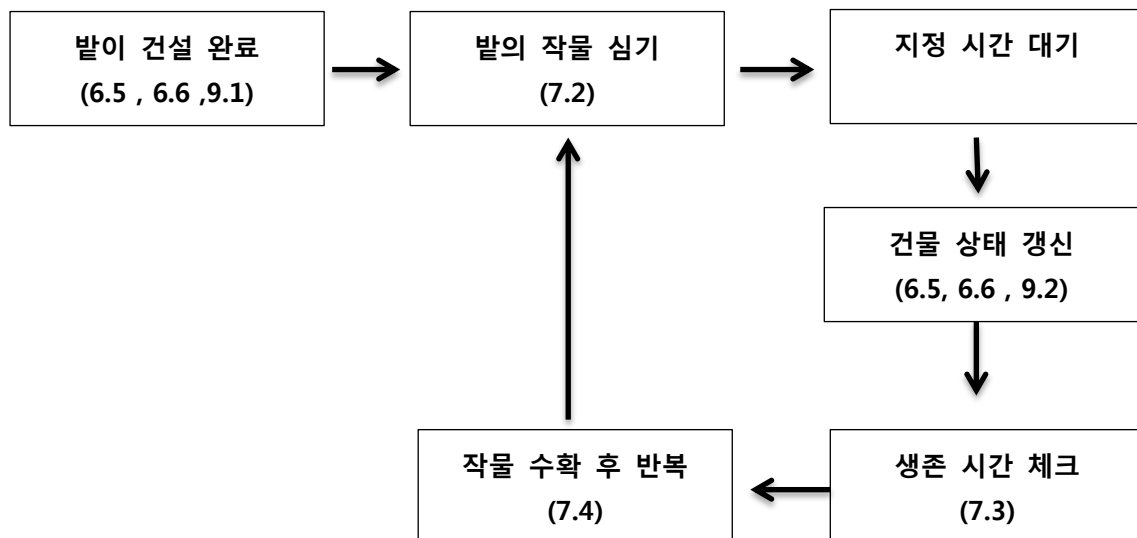
A.3 건물 건설의 흐름



A.4 건물(밭 제외)의 개인 생산 흐름



A.5 밭의 개인 생산 흐름



부록서 B: 서버 시간 얻어오기

SNG 제작 중 현재 서버의 시간을 얻어야 하는 경우가 생길 수 있다. 그 경우 다음과 같은 API를 사용하면 서버의 시간을 호출해서 얻어볼 수 있다. 이 때 호출되는 시간은 GMT+9로 서울 표준 시로 나타난다.

API: <http://swmaestros-sng.appspot.com/timeprint>

결과 반환:

`<time> YYYY-MM(월)-DD-HH-MM-SS </time>`

예제:

<http://swmaestros-sng.appspot.com/timeprint>

```
▼ <swmaestro>
  <time>2012-08-30-00-15-29</time>
</swmaestro>
```

부록서 C: 신구조문 대비문 표

0.9 버전	1.1 버전
<p>1. API 테스트 페이지</p> <p>- 해당 사항 없음</p> <p>- 해당 사항 없음</p> <p>- 해당 사항 없음</p>	<p>1. 시작하기에 앞서</p> <ul style="list-style-type: none">- 신설(기존 1장이 하위 메뉴로 들어옴) <p>9. 건물의 State 업그레이드 및 State 확인하기</p> <ul style="list-style-type: none">- 신설 <p>- 부록서 A, C</p> <ul style="list-style-type: none">- 신설 <p>- 표지 및 목차</p> <ul style="list-style-type: none">- 신설

