

OpenAPI를 활용한 Spring Framework 기반의 사용자 반응형

문화관광 시스템 개발

조규철*, 박규민^o

^o인하공업전문대학 컴퓨터정보과,

*인하공업전문대학 컴퓨터정보과

e-mail: kccho@inhac.ac.kr*, ls_al@kakao.com^o

Development of User Responsive Culture Tourism System based on Spring Framework Using OpenAPI

Kyu-Cheol Cho*, Kyu-Min Park^o

^oDepartment of Computer Science, Inha Technical College,

*Department of Computer Science, Inha Technical College

● 요약 ●

본 연구에서는 사용자의 시간·공간적 정보를 사용한 맞춤형 문화관광 시스템을 제안한다. 이 시스템은 사용자의 시스템 접근 위치와 시간을 사용하여 사용자와 가까운 문화관광지와 참여가능한 축제와 활동을 선별한다. 또한 이 시스템은 사용자의 요구에 따라 전체적인 활동을 보여줌에 있어서 OpenAPI에서 데이터를 가져와서 Chart와 Grid layout를 통해 시각적으로 이해하기 쉽게 하고, Index을 통해 정보를 분류하여 파악할 수 있게 한다. 본 연구는 정형화된 문화관광 시스템보다 다양한 형태의 문화관광이 이루어질 수 있는 시스템을 개발하였다.

키워드: Spring Framework, API, Chart, Grid layout

I. 서론

국내여행은 통계청에서 전체를 숙박여행과 당일치기 여행으로 구분하였을 때, 2018년에 당일치기 여행은 약 47%로 많은 비율을 차지하고 있다. 해마다 국내여행의 횟수가 늘어남에 따라 문화관광 시장은 점층적으로 확대될 것으로 예상된다. [1]

현재의 문화관광 시스템은 여러 여행사를 비롯한 시·도청에서 운영하고 있는데, 이는 누적된 데이터와 대표적인 문화관광 코스를 이용하여 서비스를 구축하였다. 이 시스템은 여러 기업들과 관광업자들과 협력하여 안정적인 문화관광을 할 수 있으나, 너무 정형적이고 틀에 갇혀있는 관광으로 사용자의 시간적, 공간적

정보와 무관한 시스템이다. 또한 숙박여행에 치중된 문화관광 소개로 부족한 면이 존재한다.

본 연구는 MVC 모델로 시스템을 구축하기 위해 Spring Framework를 활용하여 개발환경을 구축하였고, OpenAPI를 활용하여 문화관광 정보를 가져온다. 또한 이 시스템은 사용자의 시스템 접근 위치정보와 시간정보를 통해 사용자와 가까운 문화관광을 선별하여 보여줌으로서 당일치기 여행과 여러 문화활동을 중용하여 사용자의 의사 결정에 도움을 줄 것이다.

본 연구구성은 다음과 같다. 2장에서는 배경이 되는 현재의 문화관광 시스템에 대하여 기술하고, 3장에서는 사용자 맞춤형 문화관광 시스템을 소개한다.

4장에서는 사용자 맞춤형 문화관광 시스템의 결과를 논하고, 마지막으로 결론을 맺는다.

II. 핵심용어

1. MVC

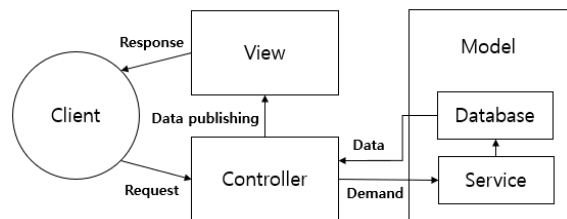


Fig. 1. MVC Pattern [2]

MVC란 Model View Controller의 약자로 그림 2와 같이 시스템을 Model, View, Controller 이 세가지 기능으로 구분한 개발 방법론이다. 사용자가 Controller에 요청하면 Controller는 Model에 요구를 통해 데이터를 가져오고, 그 데이터를 Controller를 통해 View에 넘긴다. View는 데이터 제어를 통해 사용자에게 데이터를 표현하는 역할을 하게 된다.

MVC Pattern은 사용자와 개발자가 조작 할 수 있는 영역을 구분하고 개발자가 용이하게 유지보수 할 수 있도록 하는 방법이며 Spring Framework는 대표적인 MVC Pattern의 사용 예시이다.

2. Grid layout

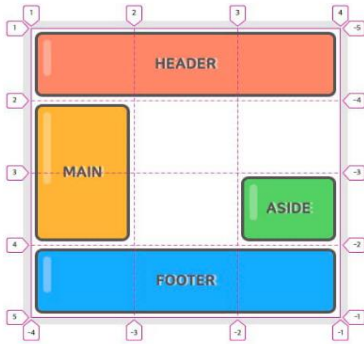


Fig. 2. Grid layout Structure [3]

Grid layout은 그림 2과 같이 웹 페이지를 layout하는 방식 중 하나이다. 웹 페이지를 구성할 때, 페이지를 세로로 일정 크기만큼 나누어 영역을 구분한다. 나눈 영역을 사용하여 블록을 맞추듯이 layout을 구성하는 것을 Grid layout이라고 한다.

Grid layout은 여러 방법론이 존재하는데 대표적으로는 CSS grid, Flex Box, Float, Bootstrap등이 사용되고 있다.

III. 관련사례

현재 문화관광 시스템은 현재 다양한 서비스를 제공하고 있다. 여러가지 테마를 통한 여행과 관광 위치와 방식에 맞게 분류하여 제시하는 서비스들이 존재한다. 이에 맞추어 쇼핑과 음식, 숙박업계와 협력하여 간편하게 해당 정보를 얻고 이용할 수 있게 하였으며, 문화관광 해설사와 예약관광까지 신청할 수 있게 되어있다. 하지만 사용자의 정보와 무관하여 사용자가 문화관광 시스템에 맞추어야 한다. 또한 숙박여행에 특화되어 있기에 이미 정형화된 코스 추천과, 대표적인 문화관광만 추천만이 가능하다. [4][5]

본 연구에서는 이러한 문화관광 시스템을 개선하기 위해서 사용자 중심의 시스템을 구축하였다. 숙박여행보다는 당일치기여행에 중점을 두고 코스가 아닌, 잠깐씩도 가능한 문화관광도 추천한다. 대표적인 문화관광뿐만 아니라 다양한 문화관광지와 문화예술활동들을 추천한다.

IV. Open API를 활용한 Spring Framework 기반의 사용자 반응형 문화관광 시스템

본 연구의 개발을 위한 환경구성은 다음 표 1과 같다. OpenAPI를 활용한 Spring Framework기반의 사용자 반응형 문화관광 시스템 개발은 다음과 같이 진행하였다.

Table 1. System Development Environment

OS	Windows 10 Pro 64Bit
IDE	Spring Tool Suite 4.3.2
Server	Apache Tomcat 9.0.30
DBMS	MySQL 5.7.26
Test Browser	Google Chrome

Spring Framework를 기반으로 JSP를 이용한 웹페이지를 개발한다.

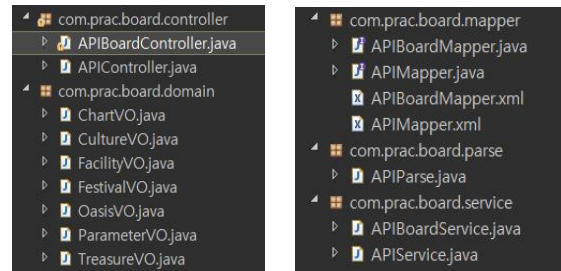


Fig. 3. Component of Spring Framework

그림 3과 같이 개발환경을 MVC에 부합되는 환경으로 구성하였다. Controller는 API Parsing을 위한 ApiController, View 운영을 위한 APIBoardController로 구성된다. Domain은 데이터를 담



Fig. 4. Result of the Grid screen

옮길 수 있는 VO(Value Object)들로 구성된다. Service와 Mapper는 Controller에서 사용되는 Interface를 정의한다. Parse는 OpenAPI에서 가져오는 데이터를 원하는 정보로 분류하는 역할을 하도록 구성되었다.

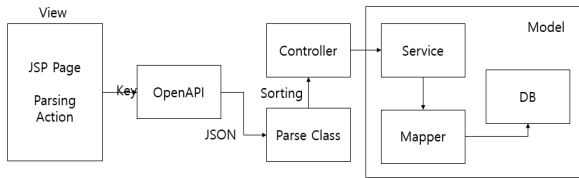


Fig. 5. The Process of entering the OpenAPI into the DB

본 연구에서 OpenAPI Parsing은 Java기반으로 설계하였으며, 정부에서 운영하는 공공데이터 포털에서 그림 5와 같이 데이터를 가져오게 된다.

JSP Page에서 Parsing Action이 일어나면 Open API에서 인증받은 Key값을 통해 필요한 API의 JSON형식 데이터에 접근한다. 접근한 JSON형식의 데이터를 Parse Java Class를 통해 필요한 데이터만 분류한다. Parse Java Class에서 분류한 데이터를 컨트롤러로 넘겨 모델에 접근한다. 모델에서는 Mapper를 이용하여 Service가 DB에 접근한다. 접근한 Service를 통하여 Parse에서 얻은 필요한 데이터를 DB에 저장하게 된다.

OpenAPI는 인천문화예술정보 공공API센터에서 관리하는 문화예술행사API, 문화기반시설API, 문화재API, 지역축제API, 문화오아시스API를 Parsing하고있다. [6][7][8]

사용자의 시스템 접근시간은 Java의 Date객체로 가져오게 되며 이는 시스템 접근시간이 된다. 사용자는 기본적으로 접근시간에 맞춰진 정보를 확인하게 된다. 확인이후에는 자유롭게 날짜를 설정하여 문화관광 정보를 찾을 수 있게 한다.

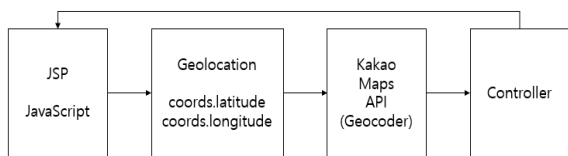


Fig. 6. The Process of Geocoding with Kakao Maps API

사용자의 시스템 접근위치는 그림 6와 같이 JavaScript와 Kakao Maps API를 이용한다. JavaScript로 사용자 IP를 통해 가져온 Coordinate 좌표계를 Kakao Developers의 Kakao Maps API를 Geocoder로 활용하여 위치를 확인한다. 확인한 위치를 컨트롤러를 통해 View로 다시 넘겨 데이터를 표현한다.

문화관광은 문화재와 대표할만한 장소에 국한된 것이 아니다. 문화관광은 단순히 문화재와 대표할만한 장소만이 아닌 여러 문화시설과 문화축제, 그리고 문화예술행사까지 다양한 요소를 포함하는 개념이다.

이를 적용한 시스템을 개발하고자 한다.

본 연구에서 문화관광 정보는 그림 4와 같이Bootstrap을 이용한 Carousel 기능을 사용하였다. 표현되는 문화관광 정보는 시스템에

접근한 사용자의 위치정보와 시간정보를 활용하여 선별하였다.

본 연구는 사용자가 시스템에 접근했을 때, 사용자가 제출한 시간 · 공간적 정보를 활용한다. 사용자가 있는 지역과 날짜를 기점으로 Grid layout을 적용하여 문화재, 문화축제, 문화시설, 문화예술 행사, 문화 특수지구인 문화 오아시스를 추천해준다.

본 시스템에서는 Grid layout을 사용하여 구성요소를 영역에 맞게 넣어 위치를 고정시켰다. 또한 사용자의 해상도에 맞게 표현되는 정보가 누락되지 않게 설정하였다.

사용자는 추천 받은 각 요소에 접근하게되면 상세한 정보를 얻을 수 있게 된다. 정보를 사용자의 환경에 맞출수 있도록 Grid layout을 적용하였으며, 요소 별로 시각적 자료인 차트를 표현하기도 하고, 색인 가능한 테이블을 배치하여 사용자가 정보를 찾을 때 불편함이 없도록 한다.

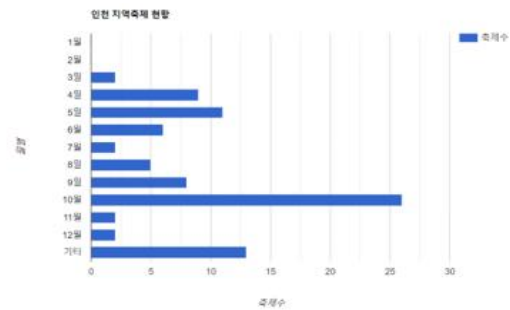


Fig. 7. Result of Chart screen with data inserted

본 시스템에서는 그림 7과 같이 Google Chart를 제공하여 여러 문화관광 정보를 사용자에게 Chart를 통하여 사용자가 이해하기 쉽게 구현하였다.

본 시스템을 사용했을 시, 사용자는 현재 위치에서 쉽게 접근가능한 문화관광지를 추천받아 잠깐의 여유나 휴식시간을 바로 즐길 수 있는 문화관광과 문화활동으로 보낼 수 있다.

사용자가 시스템에 접근하는 매개체마다 표현하는 정보의 차이가 없도록 하기위해서 그리드를 적용하였다. 그리드를 통하여 사용자는 시스템 접근 매개체에 따라 화면을 다르게 표현하며 정보 습득에 지장이 없게 하였다.

IV. 결론

현재 문화관광 시스템은 다양한 테마의 문화관광과 문화관광 코스 추천, 그리고 여러 기업과 관광업자들과의 협력으로 안정적인 문화관광을 소개하고 있다.그러나 현재 문화관광 시스템의 문제점은 정형화된 코스 추천과 대표적인 문화관광 코스만 추천하고 있고 숙박여행에 치중되었는 점이다.

사용자 반응형 문화관광 시스템을 개발함으로써 문화관광 시스템은 사용자가 가까운 위치의 문화관광을 통해 여행이라는 단어가 주는 시간 · 공간적 느낌을 탈피하여 사용자의 가까이에서도 편리하게 문화관광을 통해 위인과 휴식을 얻을 수 있는 기회를 제공하고, 사용자의 다양한 문화활동에 도움을 줄 것이다.

REFERENCES

- [1] Statistics Korea "<https://kostat.go.kr>"
- [2] CloudStudying "<https://cloudstudying.kr>"
- [3] Heropy Tech "<https://heropy.blog/2019/08/17/css-grid>"
- [4] Itour, "<http://itour.incheon.go.kr>"
- [5] Incheon tourism organization, "https://travelicn.or.kr/open_content/"
- [6] Incheon culture art information Open API center "<http://iq.ifac.or.kr/openAPI/look/index.php>"
- [7] Incheon metropolitan city open data "<https://icloud.incheon.go.kr/>"
- [8] OpenAPI, "<https://www.data.go.kr>"
- [9] Gyu-tae Chae, "Spring Boot QuickStart", pp 69~392 , June 2019
- [10] Grid layout, "<https://www.alvipixels.co.uk>"
- [11] Outsider's dev story "<https://blog.outsider.ne.kr/729>"
- [12] Springmvc "<http://egloos.zum.com/springmvc/v/504151>"
- [13] Taetaetae, "<https://taetaetae.github.io/2019/04/21/spring-boot-mybatis-mysql-xml/>"
- [14] 109.dev "<https://private.tistory.com/>"
- [15] Gimangyeol "https://blog.naver.com/khg27_200"
- [16] Slowalk "<https://slowalk.com/2270>"
- [17] Bootstrap, <https://getbootstrap.com>
- [18] DataTables "<https://datatables.net>"
- [19] Corporation NTT Data, "Spring, a thorough introduction", pp 129~327, March 2018
- [20] Geocoder, "<https://developers.kakao.com>"