

RESTful

1. REST란?

- REpresentational State Transfer

http 통신 프로토콜의 상태를 기반으로 모든 요청(Request)을 변수와값(Parameters)이 아닌 자원(URI)으로 처리한다

기존방식

<http://naver.com/Article.jsp?articleId=379>

파일명?파라미터

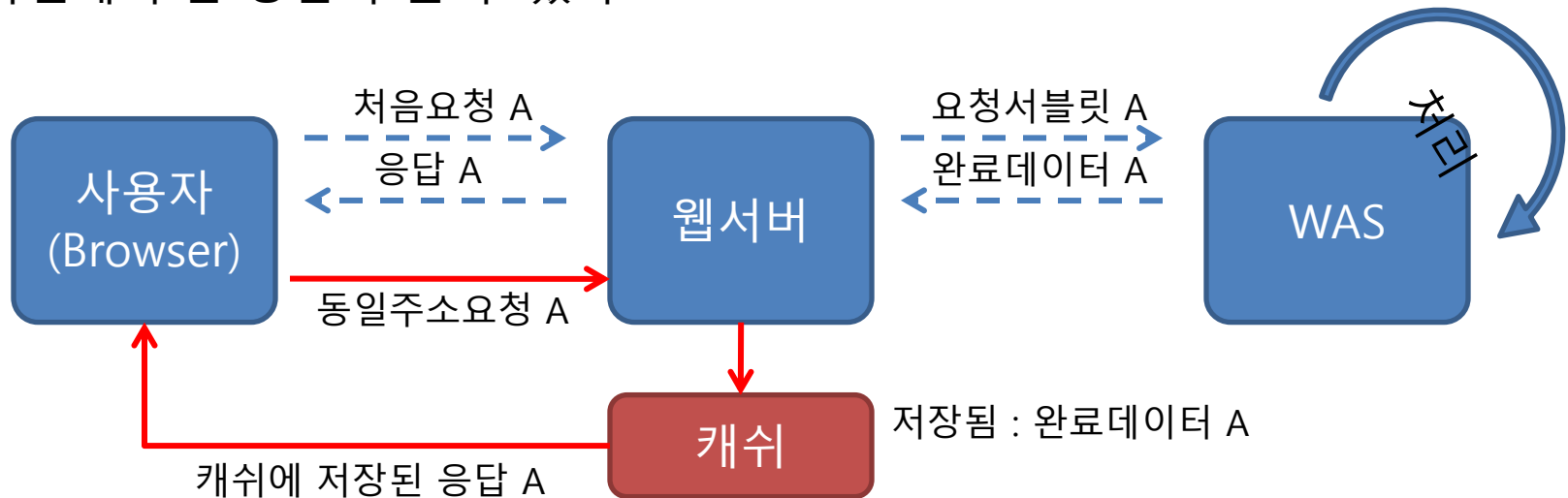
RESTful 방식

<http://naver.com/Article/379>

주소(URI)

2. REST의 장점

- 기존의 웹 인프라를 그대로 사용할 수 있고 프로토콜의 재정의가 불필요하다
- 따라서 기존 서비스를 유지한채로 적용할 수 있기 때문에 각종 네트워크, 서버 장비들을 그대로 사용할 수 있다
- 모든 서버자원을 URI로 표현하기 때문에 웹 캐쉬에 보관할 수 있고 조회(GET) 방식(Method)의 요청은 보관된 캐쉬에 의해서 서버상의 비즈니스 로직(Transaction)을 거치지 않고, 바로 서비스할 수 있기 때문에, 성능향상 측면에서 큰 장점이 될 수 있다



3. RESTful

HTTP Actions	HTTP Verbs	Path 예제	설명
index	get	/users	모든 유저 목록을 표시합니다.
show	get	/users/:id	해당 유저(:id 대신에 아이디를 입력)를 한 명에 대한 상세 정보를 표시합니다.
new	get	/users/new	유저를 생성할 수 있는 form이 있는 페이지(회원가입페이지)를 표시합니다.
create	post	/users	form의 정보를 받아 유저를 생성합니다.
edit	get	/users/:id/edit	해당 유저 정보를 수정할 수 있는 form이 있는 페이지(회원정보변경페이지)를 표시합니다.
update	put	/users/:id	form의 정보를 받아 해당 유저 정보를 수정합니다.
destroy	delete	/users/:id	해당 유저를 데이터베이스에서 지웁니다.

4. RESTful 에 대응되는 CRUD

REST	CRUD	HTTP Verbs	Path 예제	설명
index	Read	get	/주소	리스트 조회
show	Read	get	/주소/:id	상세조회 (특정 아이템 하나의 정보)
new	Create	get	/주소/new	새 정보를 받는 폼 화면
create	Create	post	/주소	정보를 생성하고 Redirect
edit	Update	get	/주소/:id/edit	수정을 하는 폼 화면
update	Update	put	/주소/:id	정보를 수정하고 Redirect
destroy	Delete	delete	/주소/:id	정보를 삭제하고 Redirect

5. Get vs Other

http://naver.com/cafe?cafeName=code

GET [/cafe](#)

http://naver.com/cafe

POST [/cafe](#)

데이터

http://naver.com/cafe

UPDATE [/cafe](#)

데이터

OSI 7 Layer

여기는 각자...