

GAE의 DataStore에 도전하다.

GAE에 대한 기본적인 환경 세팅은 완료하고 보니 가장 큰 부담으로 남은 것이 GAE의 DataStore를 이해하는 것이었다. GAE의 DataStore는 기존의 관계형 데이터베이스가 아니라 Bigtable과 Google File System을 기반으로 구축되어 있다. [How Google App Engine Datastore Works](#) 문서를 보면 좀 더 구체적으로 이해할 수 있을 듯 하다.

Bigtable과 Google File System에 대한 경험이 없었기 때문에 한번에 모든 것을 이해하는 것은 힘들다는 생각을 하고 기본적인 기능을 사용하면서 이슈가 발생할 때마다 하나씩 해결해 나가는 방향으로 진행하는 것이 좋겠다는 결론을 내렸다. 그냥 막연하게 이런 것이구나 이해하기 보다는 사용하다보면 문제가 발생하고 문제를 해결하기 위하여 해결책을 찾아보면 이해도가 높아지리라 생각한다.

그래서 처음으로 시도한 것이 [The Datastore Java API](#) 문서를 통해 기본적인 API 사용 방법을 익혀보기로 마음 먹었다. 세팅을 하고 기본적인 테스트를 진행했는데 로컬 개발 환경이 잘 지원하고 있어서 어렵지 않게 테스트가 가능했다. GAE의 이클립스 플러그인을 활용해 개발을 진행했더니 추가적인 설치 없이 로컬 PC에서 테스트가 가능했다. 단, GAE 서버를 시작한 상태에서 테스트가 가능했다.

서버를 동작하지 않고 단위 테스트를 통해 테스트하는 것이 가능하리라 생각하고 찾아봤더니 [Local Unit Testing](#) 문서에 잘 정리되어 있어 쉽게 해결할 수 있었다. 이 문서를 통해 DataStore 뿐만 아니라 Memcache, UserService 등 모든 기능에 대한 단위 테스트가 가능하다는 것을 알 수 있다.

기본적인 개발 환경을 구축한 후 GAE의 기본 DataStore API를 통해 개발을 진행하려니 아무래도 생소하고 마음에 들지 않았다. 그래서 [Datastore Overview](#)를 분석하면서 알게된 세 개의 프레임워크에 대한 분석 작업을 시작했다. Slim3는 DataStore를 분석하기 전에 GAE를 개발하기 위한 기본 프레임워크로 분석 작업을 했는데 생각보다 마음에 들지 않는 부분이 많아서 포기했던 기억이 나 일단 대상에서 제외했다.

남은 후보는 Objectify와 TWiG 두 개인데 그 중에서 Objectify를 사용하기로 결정했다. 사용하기로 결정한 가장 큰 이유는 현재 내가 개발하고 있는 환경에서 JPA를 사용하고 있기 때문에 JPA를 지원하는 Objectify가 훨씬 쉽게 접근할 수 있으며, API 또한 마음에 들었기 때문이다. 나에게 더 중요한 것은 어느 프레임워크를 사용하느냐보다는 GAE의 Bigtable과 Google File System을 이해하는 것이 더 중요하기 때문에 나머지 부분은 이미 익숙한 기술을 사용하기로 결정했다.

그렇게 Objectify로 결정하고 하루 날을 잡아서 [IntroductionToObjectify](#) 문서의 예제를 따라해봤다. 이미 JPA에 익숙했기 때문인지 예상보다 쉽게 예제를 따라할 수 있었으며, 특별히 문제가 발생하는 부분도 없었다. 이것으로 진행하면 되겠다는 결정을 하고 기본적인 개발 환경을 마무리 지었다. 드디어 개발을 시작하기 위한 준비 작업을 마쳤다. 프로젝트를 생성하고 개발을 시작하면 된다. 개발 환경을 구축하기 위한 기반 작업을 하기 위해 몇 주 동안의 시간을 소비했지만 결코 아깝지 않았다. 다양한 시도를 해볼 수 있었으며, 그 중에서 최고는 아니지만 최선의 선택을 했다고 생각했기 때문이다.

다음은 프로젝트 명을 무엇으로 할 것인지가 남았다. 프로젝트 이름은 향후 계속해서 사용될 것이며, 도메인 명으로도 사용될 것이기 때문에 결정하기 너무 힘들다. 또한 처음 시작을 무슨 기능으로 시작할 것인지도 아직 명확하지 않다. 이에 대한 고민은 다음 이야기에서...