

Utilize outras que TinyWebDB bancos de dados com o App Inventor:

Amarrar o nó com o ColdFusion

Alan Hau
Department of Computer Science
Cornell University
Ithaca, NY
United States

alan@firstcodeacademy.com

Eu sou um instrutor de K-12 alunos na primeira Academia Código, e quando os alunos querem prototipar um novo aplicativo App Inventor, eu só tenho que torná-los uma nova tabela no meu banco de dados e algumas simples mudanças no ColdFusion (CFML) scripts para os seus procedimentos. Os principais benefícios desta abordagem incluem a facilidade de gerenciamento, segurança simplificada e o mínimo de codificação para expor a API.

NoSQL DB (TinyDB) vs. relacional DB (MySQL)

TinyWebDB é uma ótima maneira de armazenar entradas de chave / valor na nuvem, mas há vezes que um usuário quer mais privacidade e estrutura do que salvar todos os seus dados no DB pública do App Inventor. Um dos mais importante limitação é que o usuário não sabe se sua chave ou tag é única quando ela usa esse no DB. Isso é problemático quando os usuários seguem o mesmo tutorial onde as chaves compartilhar nomenclatura semelhante convenções. Uma maneira de contornar isso é instalar novas instâncias de TinyWebDBs mas este é um desperdício de recursos. Também é difícil de usar quaisquer instruções SQL sobre os dados. Seleção de uma lista ordenada é feito facilmente em SQL, mas muito difícil com a linguagem do App Inventor sozinho. Outro desafio em se conectar com o SQL é para acomodar as muitas marcas e sabores em o mercado. Talvez o usuário quer se conectar a um banco de dados existente no trabalho que não pode ser mudado. Um usuário pode tentar escrever uma API web e retornar dados JSON para os seus diferentes bancos de dados, mas tomando-se como projectos no capricho poderia ser assustador.

Problemas com Ad Hoc Solutions

Se o desenvolvedor só queria ligar a sua própria base de dados, em seguida, uma das soluções poderia ser para escrever uma API em PHP. A sintaxe também é relativamente fácil de seguir e tutoriais estão disponíveis na a teia. No entanto, como o projecto cresce, alguns dos problemas tornam-se aparentes. Logins de banco de dados são materiais sensíveis, ele não deve ser exposto como texto puro no código. Se o desenvolvedor está à procura de tutoriais on-line para um script simples, detalhes de segurança são geralmente negligenciado.

Suponha que o desenvolvedor precisa para se comunicar com vários bancos de dados. administração se torna

um pesadelo. Diferentes trechos de código em várias pastas tornar a manutenção muito difícil. Um único erro ou alteração do recurso provocaria várias alterações em todo o lugar.

Ataques de injeção SQL também são comuns com o código mal escrito e é comum para Junior desenvolvedores apenas à procura de uma prova de conceito. Isso acontece quando os utilizadores introduzem código malicioso quando comunicar-se com seu Web API para executar operações não intencionais (por exemplo, excluir uma tabela ou visualização de todos nome de usuário e senhas). Estes pontos negligenciados se tornará muito problemático quando uma violação acontece e afeta os usuário.

usando ColdFusion

Eu estou usando uma solução de código aberto oferecido pelo OpenBD (Railo também funciona bem). Uma vez que você tenha seus scripts SQL favoritos embrulhados em um procedimento armazenado, use CFML para expor o procedimento armazenado de forma segura e devolver quaisquer dados em formato JSON. A sintaxe é simples e a flexibilidade é alta.

Fora da caixa, o OpenBD e Distribuições Railo pode ser executado em Windows / Mac ou Linux com um web servidor incorporado na instalação. Há nenhuma configuração adicional ou instalações dependentes necessários para iniciar executar este serviço. ColdFusion é essencialmente um fundo serviço que é executado ao lado de seu web servidor que tem a sua própria interface de administração acessível através de uma página web. Você pode registrar os seus bancos de dados e outros web alimenta-se nesta interface. todas as senhas armazenadas são criptografados.

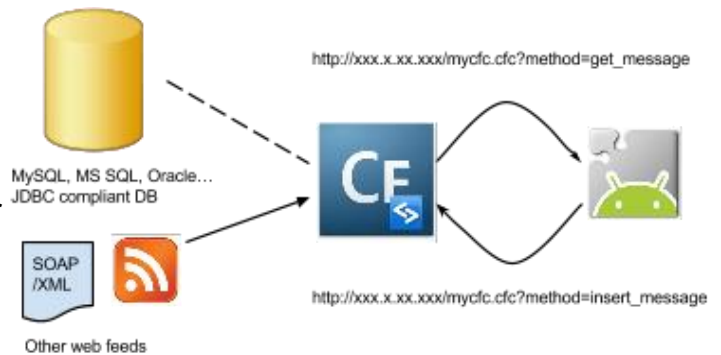


Figure 1

Uma vez que estas credenciais são armazenadas, o banco de dados será dado um tag que se refere à base de dados.

Agora eu posso seguramente compartilhar o meu código com os outros sem revelar qualquer informação de login sensíveis. Suponha que eu tem um procedimento simples armazenado escrito aqui que seleciona e classifica a idade dos alunos decrescente.

id	name	age	gender
1	Alice	12	female
2	Bob	8	male
3	Charles	15	male
NULL	NULL	NULL	NULL

Figure 2

Variations of database operations can also be wrapped in the procedure, such as INSERT, UPDATE, DELETE... etc.

```
7 CREATE DEFINER='root'@'localhost' PROCEDURE `get_agebygender` (igender varchar(45))
8 BEGIN
9     select `name`, `age`, `gender` from AJDB.StudentAge where gender = igender
10    order by age desc;
11 END
```

Figure 3

```

1 <cfcomponent>
2   <!-- cffunction name is the API method's name -->
3   <cffunction name="get_agebygender" access="remote" returnformat="json">
4
5     <!-- Specify API arguments here -->
6     <cfargument name="gender" type="string" required="true"/>
7
8     <!-- Specify Database Procedure arguments here -->
9     <cfstoredproc procedure="get_agebygender" datasource="AJDB">
10      <cfprocparam cfsqltype="cf_sql_varchar" value="#gender#">
11      <cfprocresult name="qcf">
12    </cfstoredproc>
13
14    Database reference used, no
15    credentials exposed
16
17    <!-- this is what's returned when the API is called -->
18    <cfreturn qcf>
19  </cffunction>
20 </cfcomponent>

```

Figure 4

Uma vez que este pequeno script é salva como um arquivo .cfc no servidor, ele pode ser testado imediatamente através do seu navegador favorito em qualquer dispositivo:

http://119.9.76.xxx/myapp.cfc?method=get_agebygender&gender=male

On App inventor

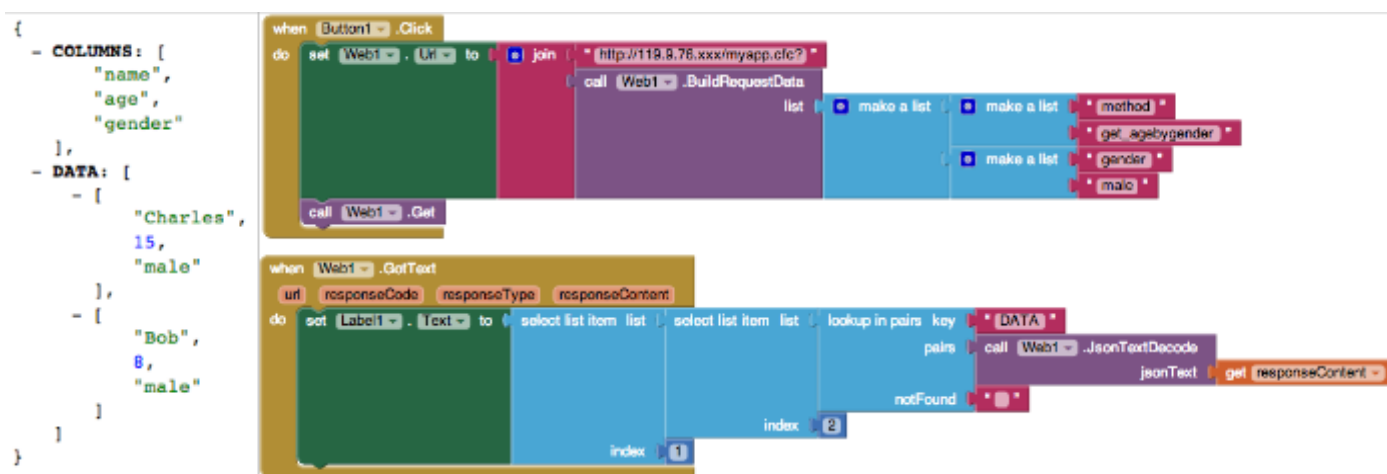


Figure 5

Este exemplo converte a cadeia de JSON para uma lista e, em seguida, exibe "Bob" em Label1.Text

Outras maneiras de usar CFML e App Inventor

Usando CFML, características App Inventor pode ser muito prolongado. Agora eu posso inserir rapidamente geolocalização de dados e até mesmo criar um cliente de chat simples. App Inventor também pode enviar e-mails através de uma chamada web API. NoSQL plugins para o MongoDB populares também está disponível para aqueles que querem uma solução de banco de dados tag base. Com segurança tratado internamente e scripts simples para se conectar com o banco de dados, os meus alunos agora passam menos tempo reinventando a roda e mais recursos de tempo criando para seus aplicativos.

