SailsJS

Generadores

- Sails trabaja con el esquema MVC.
- Sails tiene varios generadores en línea de comandos.

Generadores

Generador en línea de comandos	
sails new appNombre	Crea un carpeta llamada “appNombre” y dentro de ella genera una estructura de archivos base.
sails generate api usuario	Crea el archivo de modelo api\models\Usuario.js y el archivo de controlador api\controllers\UsuarioController.js
sails generate model usuario	Crea el archivo de modelo api\models\Usuario.js
sails generate model usuario nombre:string apellido:string	Crea el archivo de modelo api\models\Usuario.js con los atributos nombre y apellido.
sails generate controller persona	Crea el archivo de controlador api\controller\Persona.js
sails generate controller persona crear listar modificar borrar	Crea el archivo de controlador api\controller\Persona.js con las acciones “crear”, “listar”, “modificar” y “borrar”

Otros comandos en línea

Comando en línea	
sails lift	Ejecuta la aplicación creada.
sails lift --silly	Ejecuta la aplicación creada pero permite ver cada proceso ejecutado por Sails.
sails console	Esta es la consola de Sails.
sails version	Muestra la versión de Sails usada.

Archivos estáticos

- Archivos estáticos son todos aquellos archivos necesarios para un website y que están en el frontend (javascripts, imágenes, hojas de estilos, videos, etc.).
- Todos los archivos estáticos se almacenan en la carpeta assets.
- Cuando se ejecuta sails, todos los archivos estáticos son copiados a la carpeta .tmp\public.

Conexiones a bases de datos

- Sails soporta varios bases de datos como MySQL, PostgreSQL, MongoDB, Memory y Disk.
- Cada una de ellas tiene su propio adaptador que debe ser instalado.

Base de datos	Adaptador
MySQL	sails-mysql
PostgreSQL	sails-postgresql
MongoDB	sails-mongo
Memory	sails-memory
Disk	sails-disk

Conexiones a bases de datos

- Los adaptadores se instalan igual que cualquier paquete NodeJS.
- npm install sails-mysql --save

Conexiones a bases de datos

- Las cadenas de conexión se definen en el archivo config\connections.js

```
module.exports.connections = {

        <nombreConexion>: {
          adapter: <nombreAdaptador>,
          host: <direccionServidor>,
          user: <usuario>,
          password: <contraseña>,
          database: <nombreBaseDeDatos>
        }

}
```

Modelos

- Es una representación de una entidad o proceso de la base de datos
- Los modelos se almacenan en la carpeta api\models

Modelos

- Todos los modelos tiene una estructura similar.

```
module.exports = {

        connection: <nombreConexion>,
        tableName: <nombreTabla>,
        attributes: {
                <nombreAtributo>:<tipoAtributo>
        }

}
```

Modelos

- Se puede obviar el parámetro “connection” configurando una conexión por defecto en el archivo config\models.js

```
module.exports.models = {

        connection: <nombreConexion>

}
```

Modelos – Tipos de atributos

string	text	integer
float	date	datetime
boolean	binary	array
json	email	

Modelos – Tipos de atributos

```
module.exports = {

        connection: miConexion,
        tableName: miTabla,
        attributes: {
                nombre: “string”,
                apellido: “string”,
                edad: “integer”
        }
}
```

Modelos

- **defaultsTo**

```
module.exports = {

        connection: miConexion,
        tableName: miTabla,
        attributes: {
                nombre: {
                        type: “string”,
                        defaultsTo: “Sergio”
                }
        }
}
```

Modelos

- **autoIncrement**

```
module.exports = {

        connection: miConexion,
        tableName: miTabla,
        attributes: {
                idUsuario: {
                        type: “integer”,
                        autoIncrement: true
                }
        }
}
```

Modelos

- **unique**

```
module.exports = {

        connection: miConexion,
        tableName: miTabla,
        attributes: {
                correoUsuario: {
                        type: “email”,
                        unique: true
                }
        }
}
```

Modelos

- **primaryKey**

```
module.exports = {

        connection: miConexion,
        tableName: miTabla,
        attributes: {
                idUsuario: {
                        type: “integer”,
                        primaryKey: true
                }
        }
}
```

Modelos

- **enum**

```
module.exports = {

        connection: miConexion,
        tableName: miTabla,
        attributes: {
                sexo: {
                        type: “string”,
                        enum: [‘hombre’, ‘mujer’]
                }
        }
}
```

Modelos

- **size**

```
module.exports = {

        connection: miConexion,
        tableName: miTabla,
        attributes: {
                nombre: {
                        type: “string”,
                        size: 100
                }
        }
}
```

Modelos

- **columnName**

```
module.exports = {

        connection: miConexion,
        tableName: miTabla,
        attributes: {
                nombre: {
                        type: “string”,
                        columnName: “Nombre de usuario”
                }
        }
}
```

Modelos

- **required**

```
module.exports = {

        connection: miConexion,
        tableName: miTabla,
        attributes: {
                correoUsuario: {
                        type: “email”,
                        required: true
                }
        }
}
```

Asociaciones

Un usuario puede tener muchas compras. Una compra solo puede tener un usuario.

Tabla: Usuario

idUsuario	Nombre
1	Sergio
2	Mónica

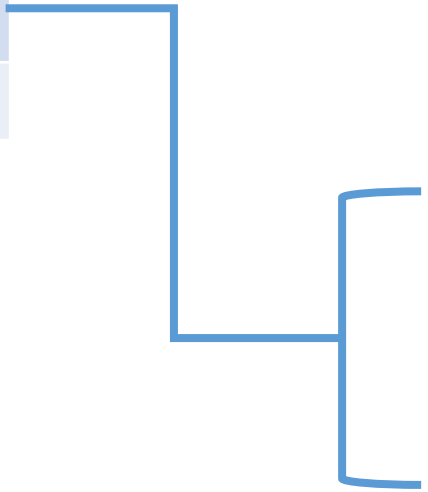

Tabla: Compra

idCompra	idUsuario	idProducto	Cantidad
1	**1**	1	10
2	**1**	5	12
3	**1**	66	4
4	2	34	6

Modelos

```
module.exports = {

        connection: miConexion,
        tableName: Usuario,
        attributes: {
                idUsuario: {
                        type: “integer”,
                        primary: true,
                        autoIncrement: true
                },
                nombre: “string”
        }
}
```

```
module.exports = {

        connection: miConexion,
        tableName: Compra,
        attributes: {
                idCompra: {
                        type: “integer”,
                        primary: true,
                        autoIncrement: true
                },
                idUsuario: {
                        model: usuario
                },
                idProducto: “integer”,
                cantidad: “integer”
        }
```

Modelos

```
module.exports = {

        connection: miConexion,
        tableName: Usuario,
        attributes: {
                idUsuario: {
                        type: “integer”,
                        primary: true,
                        autoIncrement: true
                },
                nombre: “string”,
                compras: {
                        collection: “compra”,
                        via: “idUsuario”
                }
        }
}
```

```
module.exports = {

        connection: miConexion,
        tableName: Compra,
        attributes: {
                idCompra: {
                        type: “integer”,
                        primary: true,
                        autoIncrement: true
                },
                idUsuario: {
                        model: usuario
                },
                idProducto: “integer”,
                cantidad: “integer”
        }
}
```

Asociaciones

Un dueño solo puede tener una mascota. Una mascota solo puede tener un dueño.

Tabla: Dueño

idDueno	Nombre
1	Sergio
2	Mónica

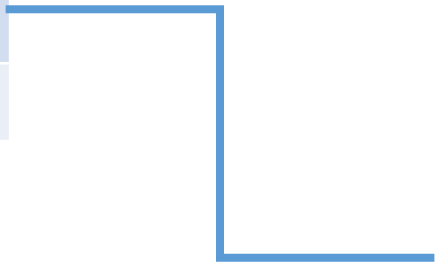

Tabla: Mascota

idMascota	idDueno	Nombre
1	**1**	Taty
2	1	Fuchi
3	1	Mandibulín
4	2	34

Modelos

```
module.exports = {

        connection: miConexion,
        tableName: Dueno,
        attributes: {
                idDueno: {
                        type: “integer”,
                        primary: true,
                        autoIncrement: true
                },
                nombre: “string”,
                idMascota: {
                        model: “mascota”
                }
        }
}
```

```
module.exports = {

        connection: miConexion,
        tableName: Mascota,
        attributes: {
                idMascota: {
                        type: “integer”,
                        primary: true,
                        autoIncrement: true
                },
                idDueno: {
                        model: dueno
                },
                nombre: “string”
        }
}
```

Asociaciones

Un persona puede tener muchas mascotas. Una mascota puede tener una persona.

```
module.exports = {

        connection: miConexion,
        tableName: Dueno,
        attributes: {
                idDueno: {
                        type: “integer”,
                        primary: true,
                        autoIncrement: true
                },
                nombre: “string”,
                idMascota: {
                        collection: “mascota”,
                        via: “idDueno”
                }
        }
}
```

```
module.exports = {

        connection: miConexion,
        tableName: Mascota,
        attributes: {
                idMascota: {
                        type: “integer”,
                        primary: true,
                        autoIncrement: true
                },
                idDueno: {
                        collection: “dueno",
                        via: “idMascota”
                },
                nombre: “string”
        }
}
```

Modelos - Validadores

```
module.exports = {
        connection: miConexion,
        tableName: Usuario,
        attributes: {
                idUsuario: {
                        type: “integer”,
                        primary: true,
                        autoIncrement: true
                },
                nombre: {
                        type: “string”,
                        minLength: 20
                },
                contrasena: {
                        type: “string”,
                        password: true
                }
        }
}
```

Modelos - Validadores

Más validadores en

http://sailsjs.org/#!/documentation/concepts/ORM/Validations.html

Modelos - Migraciones

Tipo	Acción	Comentario
safe	No modifica la base de datos	Recomendada en fase de producción
alter	Modifica la base de datos	Recomendada en fase de desarrollo.
drop	Borra la base de datos con sus registros y la crea nuevamente pero sin registros.	Usar con cuidado.

Se puede configurar la migración por defecto en config\models.js

Controladores

- Los controladores contiene la lógica del backend.
- Los controladores se almacenan en la carpeta api\controllers

Controladores

- La lógica se agrupa en acciones.

```
function(req, res) {
        res.send("Usuario registrado");
}
```

Controladores

```
//UsuarioController.js

module.exports = {

        crear: function(req, res) {
                res.send("Usuario creado");
        },

        borrar: function(req, res) {
                res.send("Usuario borrado");
        }
}
```

Rutas

- Las rutas contiene todas las rutas de acceso y los verbos permitidos.
- Las rutas se almacenan en el archivo config\routes.js

Rutas

```
'get /usuarios/': {
  controller: “Usuario",
  action: “listar"
},

‘get /usuarios/formulario’: {
    view: “formulario”
}
```

Rutas

```
'get /usuarios/listar/:pagina/:tamano': {
  controller: “Usuario",
  action: “listarPaginado"
 }
```

Políticas

- Las políticas definen qué acciones puede hacer un usuario.
- **Las políticas solo se pueden aplicar para controladores.**
- Las rutas se almacenan en la carpeta api\policies

Políticas

```
//UsuarioAutenticado.js

module.exports = function(req, res, next) {

        if(req.session.usuario!=undefined) {
                next();
        }

        res.forbidden("Usted no tiene acceso");
}
```

Políticas

```
//UsuarioAutenticado.js

module.exports = function(req, res, next) {

        if(req.session.usuario!=undefined) {
                next();
        }

        res.forbidden("Usted no tiene acceso");
}
```

Protegiendo los controladores con Políticas

- En el archivo config\policies.js se definen las políticas que se aplican a un controlador.

Protegiendo los controladores con Políticas

```
//policies.js

module.exports.policies = {

        *: false,

        UsuarioController: {
                *:false,

                listar: true,
                insertar: “UsuarioAutenticado”,
                borrar: [“UsuarioAutenticado”, “UsuarioAdministrador”]
        }
}
```