



06

## Exercício 1

Crie a classe `AlunoConverter` no pacote **`br.com.caelum.converter`** e implemente o método `toJson()`.

```
JSONStringer jsonStringer = new JSONStringer();

jsonStringer.object().key("list").array().
    object().key("aluno").array();

for (Aluno aluno : alunos) {
    // para cada aluno usar o JSONStringer
}

String json = jsonStringer.endArray()
    .endObject().endArray().endObject().toString();
```

Para cada `Aluno` precisaremos dizer ao **`JSONStringer`** como criar no **JSON** as informações do aluno:

```
jsonStringer.object()
    .key("id").value(aluno.getId())
    .key("nome").value(aluno.getNome())
    // demais informações do aluno...
    .endObject();
```

Alguns métodos lançam `Exception`, coloque todo o código dentro de um `try catch`!

Vamos ver o **JSON** gerado para os alunos cadastrados em nosso SQLite.

Customize o método `onOptionsItemSelected` da classe **`ListaAlunosActivity`**:

```
public class ListaAlunosActivity extends Activity {
    //atributos e métodos

    @Override
    public boolean onOptionsItemSelected(MenuItem item) {
        switch (item.getItemId()) {
            //restante da implementação já feita

            case R.id.menu_enviar_alunos:
                AlunoDAO dao = new AlunoDAO(this);
                List<Aluno> alunos = dao.getLista();
                dao.close();

                String json = new AlunoConverter().toJson(alunos);

                Toast.makeText(this, json, Toast.LENGTH_LONG).show();
                return false;
        }
    }
}
```

Como ficou a implementação de seu `AlunoConverter`?

## Responda

| INSERIR CÓDIGO | FORMATAÇÃO |
|----------------|------------|
|                |            |

