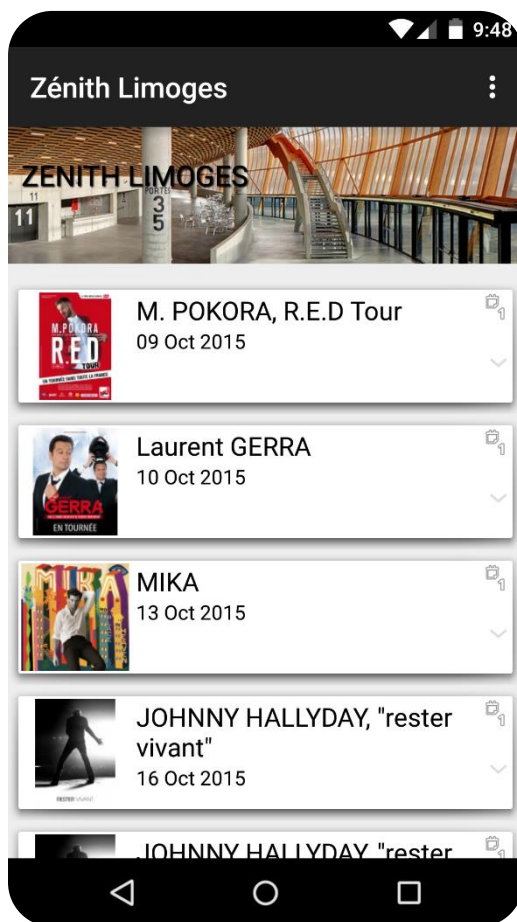


# Summer School 2015

## Mobile Application

Zénith Limoges:

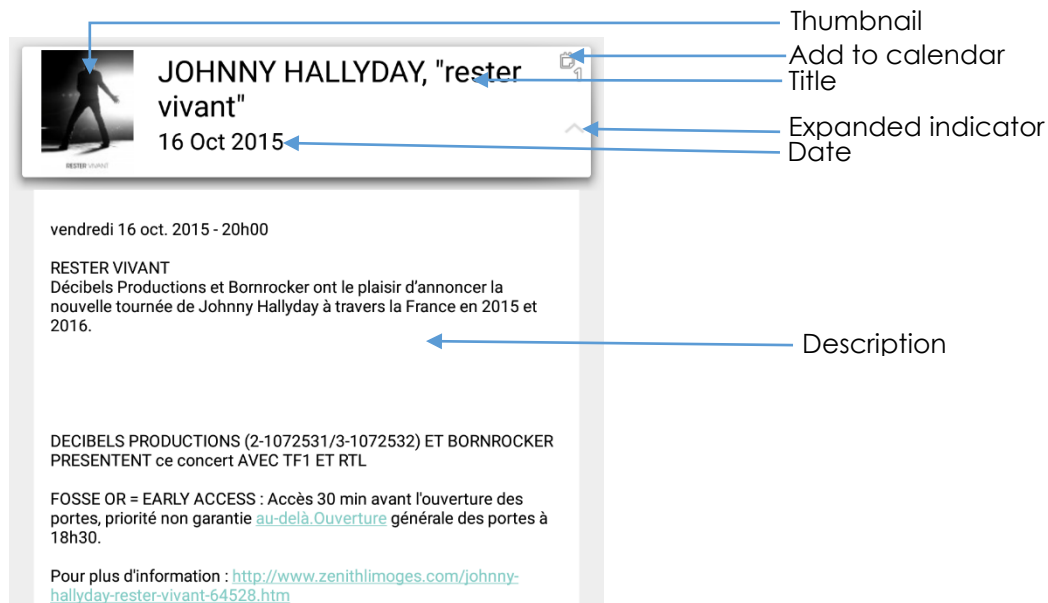


LUVISON Nicolas



## What does it consists?

The application will download a RSS feed from the web site then display the fetched data.



## The necessary

- A thumbnail
- A title
- A date
- A description

## Options

- Button add to calendar
- Clickable link and number phone
- Rotation screen don't restart the downloading data
- Launch page

## Why this application

Since I came to Limoges I often search on the Play Store an application that displays upcoming events but it never been created

So thanks to the course I decided to make one.

It is faster than go to the website and search for new events.

## The good points

Learn how to use expanded listview

Learn how to do AsyncTask

Learn a better to use thread and handler

Learn how to use XML parser.

## Problems encountered

The main part of the application that displays the event was very quick and easy to write, but many problems arrived.

The asynchronous download of thumbnails:

First I called the method from the adapter. But the adapter called the download too many times, actually each time we interact with the view. So I decide to call the method only at the instantiation of a header. But if the download was finish before the listview get its adapter, I had to add a new variable in header to store the memory address of the adapter and call the method notifySetDataChanged.

The rotation of the screen:

I ate when I turn the screen and the application restart entirely. I used onSaveInstanceState to save the data. To use that I needed to implement Serializable. The problem was that Serializable try to save the memory address of expandableLisviewAdapter which not exists anymore when the app is killed by the operating system and it lead to a bug.

XML parser:

I had a problem when parse the xml due to a bad condition, I get the previous link in the XML

Freezing startup:

At the beginning I used a loop which stop main thread while the parser was not finished. I didn't see the problem before I had a launch picture with a progreesbar. I fix the problem thanks to a handler which test each 100 millisecond if the parser has done its task.

Clickable link:

At first I tried with the xml property autoLink="true" and with the interface Linkify. This might have worked if the method was called inside the main activity. So I had to use a SpannableString and a pattern. The final problem was the fact I've didn't understand the documentation and used result.group(i) which return the parts of the regex.

## Conclusion

The application implements all the optional functionalities. I tried to resolve the bug I could found but I can't find all bugs in virtual device.

This exercise was really instructive and interesting.