

Define a customized course and import it into Moodle without changes to the configuration of the Moodle system

Dimitri Bigler

Faculty of Computer Science
University of Applied Sciences Kempten
Kempten Bavaria Germany
dimitri.bigler@hs-kempten.de

Georg Hagel

Faculty of Computer Science
University of Applied Sciences Kempten
Kempten Bavaria Germany
georg.hagel@hs-kempten.de

ABSTRACT

This paper describes an investigation into the possibility of defining a customized course in Moodle, a popular learning management system. The aim was to define a course independently of Moodle, and to implement the course without making any modifications to the configuration of the Moodle system. Two methods were considered: The Moodle REST-API and the restore functionality of Moodle. The latter method was chosen, and the paper details the structure of Moodle backup zip (MBZ) files and the creation of a generator that builds the required file and folder structure. The generator reads a Domain-Specific Language (DSL) document, parses it, and creates the required backup folders and XML files, which are then packed into an MBZ file. This paper provides technical details about the MBZ file structure and the implementation of the generator, as well as a flow chart that describes the steps needed to create an H5P-File into the MBZ file structure. This work has implications for the development of customized courses in Moodle without modifying the Moodle instance.

Permission to make digital or hard copies of part or all of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for third-party components of this work must be honored. For all other uses, contact the owner/author(s).

ECSEE'23, June 19-21, 2023, Seon/ Bavaria, Germany

© 2018 Copyright held by the owner/author(s).

ACM ISBN 978-1-4503-0000-0/18/06...\$15.00

KEYWORDS

Moodle, learning management systems, customized course, Moodle REST-API, Moodle Backup Zip (MBZ) file, Domain-Specific Language (DSL), reverse engineering, gzip and tar packed XML-files, learning elements, Moodle restore functionality, Authoring tool, Generator

ACM Reference format:

Dimitri Bigler and Georg Hagel. 2023. Define a customized course and import it into Moodle without changes to the configuration of the Moodle system. In *ECSEE'23: European Conference of Software Engineering Education 2023, June 19-21, 2023, Seon/ Bavaria, Germany*. ACM, New York, NY, USA, 5 pages.

1 Introduction

Learning management systems (LMSs) are virtual platforms that provide learners access to various learning elements for planning and managing their learning activities online and offline. These platforms enable the planning and management of individual and organizational learning processes, distribution of learning content, bundling and development of knowledge from practical projects, administration of learners, and documentation of learning outcomes. One well-known LMS is Moodle an open-source software that can be installed on any computer that can process PHP and support an SQL database. As of February 16, 2023, Moodle is used in 238 countries and over 164,800 active Moodle websites are registered [1]. Given its wide acceptance, we decided to concentrate our research and development efforts on Moodle. Our research question was whether it was possible to define a customized course and somehow create it inside of Moodle without modifying the Moodle configuration. We did not want to make any changes to the setup since many learning institutions do not allow such modifications for security and administrative reasons. The motivation behind this idea is project AdLer (Adaptive Digital Learning Spaces), a collaborative project between the Technical University of Aschaffenburg and the University of Applied Sciences Kempten. AdLer aims to improve digital learning platforms and media for students with a focus on motivation and individual competency development. The project wants to create learning paths with an authoring tool that can be exported to learning platforms and as 2D and 3D learning games. The 2023 publication, "Individual games in higher education" by Tobias Eigler et al., provides a comprehensive examination of both the authoring tool and the resulting learning games. [2]

2 Motivation

As previously mentioned, our goal is to develop a tailored course outside of the Moodle system without altering the existing configuration. Our approach offers several compelling benefits that could be of interest to universities and other educational institutions. Many common approaches, such as utilizing plugins or making modifications, require changes to the Moodle

configuration, which can result in additional administrative and cybersecurity-related work for these institutions. Additionally, many of these modifications lack necessary security patches, which could lead to potential security vulnerabilities.

In this paper, we will explore the underlying principles of the Moodle Backup file structure and demonstrate how it is possible to incorporate additional features into course creation using Moodle's Restore functionality. By avoiding the need for extensive configuration changes or modifications, our approach can streamline course development while reducing the associated cybersecurity risks.

Presently the Backup structure of the restore file remains undocumented. Documenting the underlying structure could prove beneficial for developers in a variety of contexts, whether for development purposes or to repair an existing Moodle backup file. Publicly documenting this information would enable developers to gain a deeper understanding of the underlying structure and make more informed decisions about how to approach course development and restoration. Additionally, it would help to standardize the backup file format, allowing for greater interoperability and ease of use across different Moodle instances.

3 Related Work

Numerous papers discuss the creation and adaptivity of Moodle courses. However, our aim was to generate a custom course without modifying the Moodle configuration. In a well-known paper by Sabine Graf [3], she discusses the adaptivity of Moodle courses and how it is possible to alter the course to suit the student's needs. However, this functionality requires extending the Moodle system with a self-written plugin. Other papers written by Rüdian and Pinkwart [4], Ramprakash [5] or Zabolotskikh [6] utilize frameworks, plugins, or modifications to the Moodle configuration to achieve the desired functionality. Thus, we sought an alternative solution and decided to generate our own MBZ file and use Moodle's restore functionality.

4 Technical Description

After conducting research, we identified two solutions that could fit our needs to create a customized course. The first method is through Moodle's REST-API, which provides a lot of functionality and a long list of functions. However, some course settings, such as completion conditions of activities, various theme settings (icons, colours, tiles, subtitles, and tile progress), or tags, are currently not available. Additionally, as of today it is not possible to edit an existing course through the REST-API at all. Therefore, we decided not to use this method.

The second method uses the restore functionality of Moodle. For every course backup, Moodle creates a Moodle backup zip (MBZ) file that holds all the information about the course. Since this file format is not documented, we had to do our own research on how the file is structured. The goal is to define a custom course in an authoring tool, recreate it into an MBZ file, and lastly restore it into the Moodle system. The restore function brings a big advantage compared to other approaches. Firstly, the files can be restored in

every Moodle instance worldwide. That being the case, the user that restores the course has to have the sufficient permissions.

Additionally, the restored course does have the same functionality as courses that were created inside of Moodle. Given that it is possible to include user data in the backup file, it should also be possible to "edit" a course, by editing the MBZ file and restoring it. Since this approach does not seem to result in any loss of features, we have decided to adopt it.

5 MBZ file structure

Upon closer examination, the MBZ file is a gzip and tar packed XML-file and folder structure. The various files hold all the course information. To understand what information each file holds and how the files refer to each other, we had to reverse engineer the file. We created a course with a minor change and looked at the difference between the previous and newly created folder structure. After some time and tries, we figured out how to create various learning elements and how to structure the course.

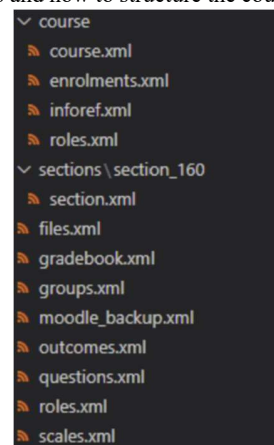


Figure 1: Mandatory XML files and folders in the MBZ file

Every MBZ file structure has mandatory files and folders representing the default settings of a Moodle course. Figure 1 illustrates all the mandatory files and folders for a MBZ file. If an incomplete MBZ file is uploaded into Moodle, the system will give an unspecific error message and abort the restore process.

The **course** folder holds all the information about the course settings. That includes settings, such as enrolment method, course name, course format, start and end date of the course, etc.

The **sections** folder represents the topics inside of Moodle. Every topic gets its own folder and file. The file holds information about the topic name and id-number. To place the learning elements in the correct topic, they refer to this id.

The **moodle_backup.xml**-file holds a lot of information regarding the course elements, topics, and their position in the course. This file combines the information from the other folders and files and acts as an overview file for the restore process.

All the other files hold additional information or are empty, nonetheless they are mandatory for the restore process. Without them the course restore cannot be completed.

To document the MBZ file structure, we created flow charts for the different learning elements and course settings. Figures 2 and 3 illustrate a section of a flow chart, that describes, how to add an H5P Learning Element into the MBZ file structure. Once all steps of the chart have been completed, the file structure can be packed as a tar and then compressed as a gzip. The resulting file has an identical structure to an MBZ file and can be restored to the desired Moodle instance.

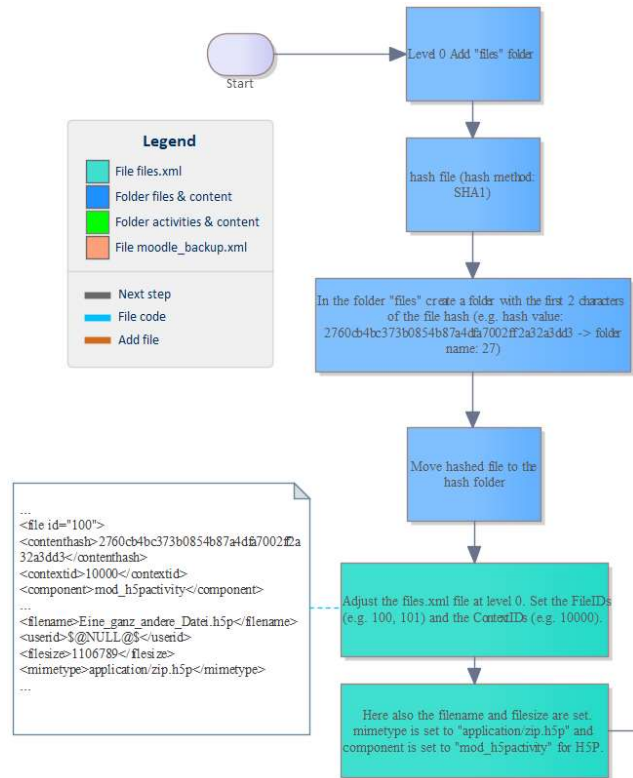


Figure 2: Add an H5P Element to the MBZ file structure Flow Chart – Part 1

In Figure 2, we illustrate the necessary steps to add an H5P Element to the MBZ file structure. The process begins with creating the "files" folder at the root of the structure, and then proceeds through a series of steps to ensure that all required files and folders are edited and created. These steps are crucial for properly adding the H5P Element to a Moodle course and ensuring its correct display.

All flow charts are based on an empty course MBZ file, which contains all the essential information needed for a course. Due to space constraints, we present only the starting and ending sequences of the flow chart in this paper. However, a complete version of the chart, as well as other full-size charts depicting different operations, such as adding various learning Elements and editing course settings, can be found on our GitHub repository [7].

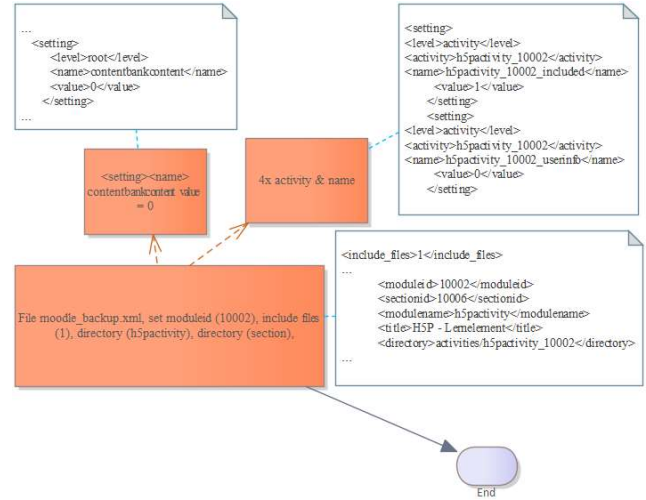


Figure 3: Add an H5P Element to the MBZ file structure Flow Chart – Part 2

Figure 3 depicts the final steps required to include an H5P Element in the MBZ file. These steps involve making changes to the moodle_backup.xml file, which stores all the critical information about a course's elements and settings.

6 Development

As soon as we understood the basics of the MBZ file structure we could start with the implementation of a generator that creates the desired files and folders. To create a customized course outside of Moodle the author creates said course in an authoring tool.

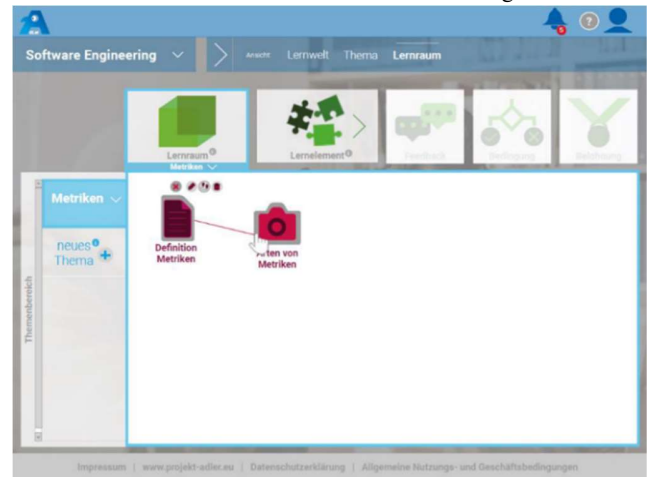


Figure 4: Authoring tool customizing course and learning elements prototype

Figure 4 displays a prototype of the authoring tool. In this illustration the author describes the course structure, learning elements and restrictions to unlock elements. The user can create a course using an interactive interface, save it, and generate an MBZ

file from it. When the author finished his custom course, the generator starts to produce the required files and folders. The generator reads the status of the learning world entity, creates a Domain-Specific Language (DSL) document, parses it, and creates the required backup folders and XML-files, which are then packed into a MBZ file.

To create the correct MBZ file structure, various classes and attributes must be nested within one another and then serialized. These classes are called entities and several are needed to correctly represent the structure of a file. To generate learning resources such as topics, labels, and learning elements, a total of 122 classes and interfaces are required. To serialize the necessary XML files with the appropriate contents, the relevant classes are initialized and serialized in their respective factories. Finally, the BackupFileGenerator class is responsible for creating the entire Moodle backup structure.

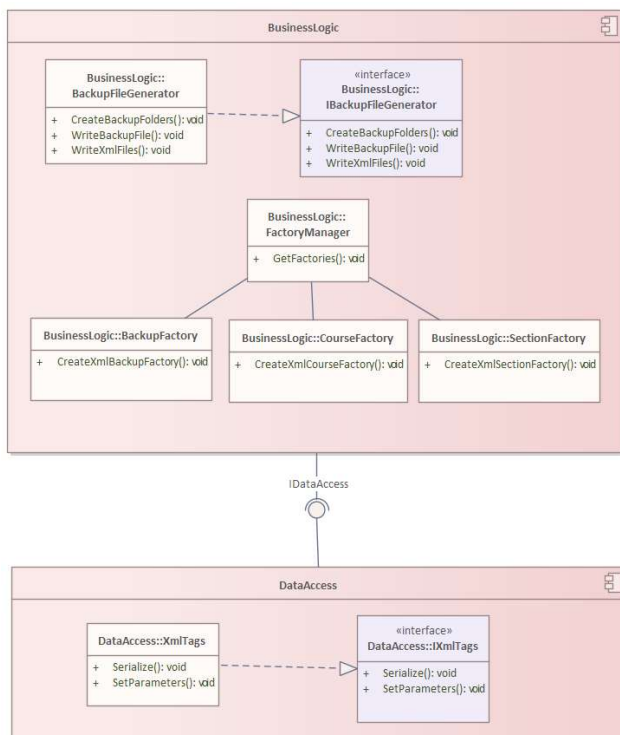


Figure 5: Generator architecture

The architecture of the generator is illustrated in Figure 5. The BackupFileGenerator class is the starting point, responsible for creating the required directory structure and invoking all associated factories. Each major directory and file possesses its own dedicated factory that generates the required entities and configures their attributes. Once the attributes have been established, the factories serialize the objects and store them in their intended destination. Finally, the BackupFileGenerator class collates the root folder and compresses it using tar and gzip. The resulting output adheres to

the structure of an MBZ file and is suitable for uploading to a specified Moodle instance. The illustrated architecture of the generator is currently expanding, and as such, additional factory classes will inevitably be required in the future to adequately encapsulate all of the functionalities of Moodle.

7 Limitation

As our aim was to create a custom course without modifying the Moodle instance, our course customization is restricted by the Moodle's existing functionality. Thus, we can only use Moodle activities and course settings that already exist. However, this is adequate to create a custom course with all the desired learning elements and paths (with the help of Moodle restrictions). Lastly the MBZ file can fully support all the course creation functionality and capabilities offered by Moodle.

8 Results

Presently, the authoring tool and generator can create a course with learning elements (e.g. pdf, h5p, YouTube links, files) and topics. Additionally, it is possible to customize the course name, enrolment method, and theme. In the future, we intend to implement restrictions between the learning elements to be able to represent learning paths.

9 Conclusion

Our approach is very uncommon, as the restore function eliminates the need to modify the Moodle instance, making it more accessible to other potential users.

One of the significant challenges associated with this approach is that the MBZ file is not documented, and any functionality that we want to implement in the generator requires reverse engineering of the file. After restoring the resulting file, it may be incomplete, or some attribute may have been set incorrectly, resulting in Moodle system errors. This makes the entire process tedious and time-consuming.

ACKNOWLEDGMENTS

This work was supported by the German Federal Ministry of Education and Research (BMBF), grant number 16DHBKI036, and the Bavarian State Ministry of Science and Art. The responsibility for the content of this publication remains with the authors.

REFERENCES

- [1] Moodle, „Moodle,“ [Online]. Available: <https://stats.moodle.org/sites/>.
- [2] T. Eigler, P. Bartel, A. Weishaupt, G. Hagel, M. Klopp, A. Döringer, M. Hochstetter, J. Abke, J. Elsebach, R. Rossmann, „Individual games in higher education,“ *ECSEE 2023*, 2023.
- [3] S. Graf, „Providing Adaptive Courses in Learning Management Systems with respect to Learning Styles,“ p. 8, 2007.
- [4] S. Rüdian and N. Pinkwart, „Generating adaptive and personalized language learning online courses in Moodle with individual learning paths using templates,“ <https://ieeexplore.ieee.org/document/9499916/>, pp. 53-55, 2021.
- [5] P. Ramprakash, S. Saravanakumar, M. Veeraprakash, S. V. Prasanth und M. Y. Asarudheen, „Customized learning path using Moodle,“ *International*

Journal For Technological Research In Engineering Volume 3, Issue 6, p. 4, 2016.

- [6] A. Zabolotskikh, A. Zabolotskikh, T. Dugina und D. Tavberidze, „Creating individual learning paths in the Moodle plugin for undergraduate students to study English grammar,“ *Educ Inf Technol (Education and Information Technologies)*, pp. 617-637, 2020.

[7] Projekt AdLer, „Github,“ [Online]. Available: <https://github.com/ProjektAdLer/Paper>