

Business Layer Logic (BLL) is about data?

Asked 10 years, 1 month ago Active 10 years, 1 month ago Viewed 4k times



7

I think BLL is about Data. It should not include a method called SendEmail. BLL is a place for caching data, manipulating it, doing calculations related to business. Sending email is a business process but the code which actually send the email should be outside of BLL namespace.



Is BLL only about data ?



4

[architecture](#)[business-logic](#)

edited Jan 5 '10 at 15:54

[cgp](#)

37.7k

10

93

129

asked Jan 5 '10 at 15:51

[Costa](#)

3,541

10

41

72

Unless your business is sending email. – [cgp](#) Jan 5 '10 at 15:57

@alt - Even if his business wants to send email, the definition of how to send an email shouldn't be defined inside of a BLL. It should be separated into a Utility class. – [JonH](#) Jan 5 '10 at 15:59

6 Answers



13

BLL is not about data, It is all about what needs to be done with the data.



- User will only interact with the front-end presentation-forms of any application, which is popularly known as **presentation layer**.
- Data will be displayed or exchanged as input/output to this layer from various **sources of data**. These sources are **databases** or **web-services**. The piece of code that actually fetches or sends these data to respective sources of data is what we call as **DAL** - data access layer.
- In between the application does special operations which we call **application-requirements** or **user-needs**. This strategic part of the application is called **BLL** that actually **addresses needs of the client** for which you are developing the application.
- If data needs to be stored in the database, **BLL will have DAL as an underlying layer**.

- **BLL is not aware of the sources of data and how data is fetched or sent** to data-sources. You have DAL for that. BLL only knows about data, that to in forms of business-objects more often and operations on the **data business-objects**.
- **BLL also does not know whether user is using website or a desktop application**. You have presentation layer for that.

edited Jan 7 '10 at 10:48

answered Jan 7 '10 at 10:41

[this.__curious_geek](#)

39.1k 19 102 128

- ▲
- 3
- ▼
- 🕒
- BLL stands for your business logic layer. It should handle anything related to your business logic layer. SendEmail may be better in some sort of utility class?
- Also if you tell your BLL about an emailing mechanism you are giving it too much information (tightly coupled, follow the law of demeter for functions, wiki / google it). Your BLL doesn't care about email nor should it.
- When you mentioned **Data**, you are probably after the DAL (Data Access Layer). This is the layer that deals with data inserts / updates etc back to some resource such as a database.

answered Jan 5 '10 at 15:53

[JonH](#)

30k 11 77 131

- ▲
- 2
- ▼
- 🕒
- BLL isn't about data...it's about Business (hence **Business** Logic Layer).
- The DAL is all about Data.
- I'd probably move the SendMail method to a Utility class, but it's perfectly legitimate that you'd need to call SendMail from the BLL.

answered Jan 5 '10 at 15:55

[Justin Niessner](#)

220k 32 375 511

Thanks, What if you have a method called SendEmail in BLL the method will actually do several things 1- Using utility class called Util it will encrypt and send the email. 2- Insert the sending status into the database through DAL. – [Costa](#) Jan 5 '10 at 18:48

In that case, I would say the BLL is the correct location for the method...although the name could be a little more descriptive (to distinguish itself from a Utility style SendEmail method). – [Justin Niessner](#) Jan 5 '10 at 19:18



If you look at it from a layer perspective, sending emails would fit better into the presentation layer rather than the business logic or data layer.

0



However, the triggering of sending an email may come from the business layer, and the business layer should not be calling the presentation layer.



In which case a potential solution would be for the business layer to manage an email queue, and have the presentation layer manage picking up the emails and sending them.

Sometimes conforming rigidly to a pattern can cause more problems than it is trying to solve. If you find a specific implementation works for you now and won't cause any problems in the short to medium term, and the cost of investigating and implementing the "perfect" solution is too great, then go with what you have got.

answered Jan 7 '10 at 10:15



cjk

41.6k

8

71

107



0



The Business Layer should contain classes that hold business information. Classes in this layer should represent your business in software. Methods should involve business rules. The business layer will hold, validate, and manipulate data, but your underlying Data Access Layer (DAL) will know how to add, remove, get, and update data from the database. The business layer shouldn't care about presentation either.



In the past teams I've always worked on separate functions that can appear in any program/business like sending a email in it's own generic class/method. The only time I've seen a BLL class have any ties to a email is when the business rule was written to send a email. In this case the BLL knew the text of the email to send but instantiated the general email class to send the email.

answered Jan 7 '10 at 11:00



Jason Too Cool Webs

108

8



Your business logic layer should handle things relating to your business, to say that your business layer is only about data would not quite be accurate. For example many people have the requirement to cache data for performance reasons, and so to say that caching is

0

business specific would be incorrect.



Certain calculations however (for example calculating a quote) can definitely counted as business logic, as they are specific only to your business.



Sending email is definitely not business logic - this is a fairly generic requirement and certainly isn't specific to your business or industry.

answered Jan 5 '10 at 16:05



Justin

76.5k

43

199

335