

Proto Buffer

Document share application

Summary

- Title
- Theme
- Scope
- Tools
- Motivation
- Screens
- Classes

- Protocol Buffer
- Google
- 2001

Title

Proto Buffer

Theme

Document share

• Books

• Comics

Mangas

Magazines

Scope

- Secure
- Free
- Fast

Programming language C++

Multiple paradigms, fast and has an efficient development

Editor Visual Studio Code

Multiple extensions that enable a Full Stack development environment

Platform Linux

Widely adopted by developers and has a concise base to development in

Database Mongo

Document oriented, flexible, scalable and consistency

GUI

QT, Nuklear, Imgui or Nana

From an Anarcho-capitalism perspective, the intellectual property doesn't exist, because intellectual property its not scarce. One person can read a document file at the same time that another person reads the same document file, but a person can't use a physical object at the same time that another person, because this goes against the physic laws.

The state architecture monopolizes all the things with autoral rights and patents, but this doesn't cause the success of authors. Their hard work and value perception of the market by its consumers is what truly gives 'value' to the product and return profit to the authors.

An user can download every existent TV show via Torrent, but the majority people choose to pay for services like Netflix, HBO GO, Hulu, Amazon Prime, etc; because they see the practicality in these services. The same goes to the Kindle, Google Books, Packt, Humble Bundle and other book services. If no one pays for the books of an author, this author probably will not have the incentive to produce more books, but its previous productions probably do not have a visible or truly value, once no one wants to contribute to.

Some platforms like Passei Direto and Scribd, enables users to share documents, but these platforms remove the download option and restring the read of the content sent by other users if a payment is not realized. To go against this platforms its proposed.

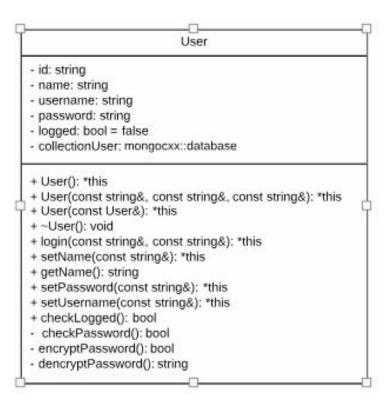
Screens

- Welcome
- User registration
- User login
- User profile update
- User document list
- Document upload
- Document search
- Document view
- Document update

```
Connection
- prefix: string = mongodb
- host: string = localhost

    port: string = 27017

- uri: mongocxx::uri
- instance: mongocxx::instance
- static client: mongocxx::client
# Connection(): *this
# Connection(const string&, const string&, const string&): *this
# Connection(const Connection&): *this
# ~Connection(): void
# getUri(): mongocxx::uri
# getCollection(const string&): mongocxx::database
# setPrefix(const string&): *this
# getPrefix(): string
# setHost(const string&): *this
# getHost(): string
# setPort(const string&): *this
# getPort(): string
# explicit operator bool() const: bool
- concatConnectionUri(): string
- updateConnection(): bool
+ ping(): bool
```



Document

- id: string
- title: string
- type: string
- idiom: string
- userld: string
- createdAt: auto
- keyWords: array
- categories: array
- collectionDocument: mongocxx::database
- + Document(): *this
- + Document(const string&, const string&, const string&, const array&,
- const array&): *this + Document(const Document&): *this
- + ~Document(): void
- + setTitle(const string&): *this
- + getTitle(): string
- + setType(const string&): *this
- + getType(): string
- + setIdiom(const string&): *this
- + getIdiom(): string
- + setUserId(const string&): *this
- + getUserId(): string
- + setCreatedAt(const auto&): *this
- + getCreatedAt(): chrono::system_clock
- + setKeyWords(const array&): *this
- + getKeyWords(): array
- + setCategories(const array&): *this
- + getCategories(): array
- + search(const string&): array
- + list(): array

Metric

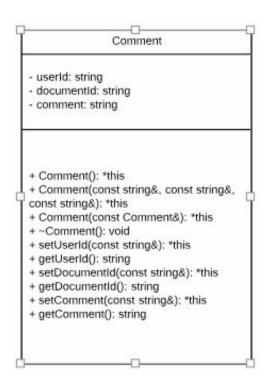
- likes: unsigned int
- views: unsigned int
- dislikes: unsigned int
- documentId: string
 collectionMetrict: mongocxx::database
- + Metric(): *this
- + Metric(const unsigned int&, const unsigned int&, const unsigned int&, const string&): *this
- + Metric(const Metrict&): *this
- + ~Metric(): void
- + setLikes(const unsigned int&): *this
- + getLikes(): unsigned int
- + setViews(const unsigned int&): *this
- + getViews(): unsigned int
- + setDislikes(const unsigned int): *this
- + getDislikes(): unsigned int
- + setDocumentId(const string&): *this
- + getDocumentId(): string

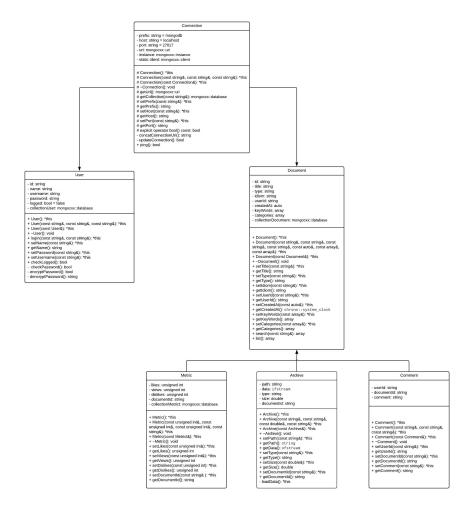
Archive

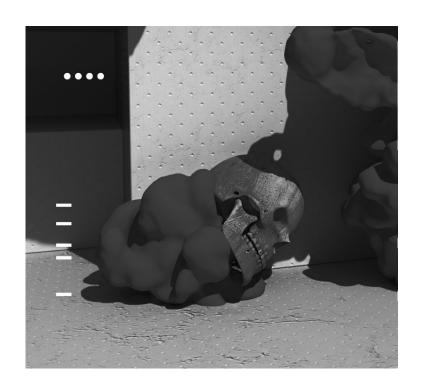
- path: string
- data; ifstream
- type: string
- size: double
- documentId: string
- + Archive(): *this
- + Archive(const string&, const string&, const double&, const string&): *this
- + Archive(const Archive&): *this
- + ~Archive(): void
- + setPath(const string&): *this
- + getPath(): string
- + getData(): ofstream
- + setType(const string&): *this
- + getType(): string
- + setSize(const double&); *this + getSize(); double
- + setDocumentId(const string&): *this
- setDocumentia(const si
- + getDocumentId(); string
- loadData(): *this

Archive

- path: string
- data; ifstream
- type: string
- size: double
- documentId: string
- + Archive(): *this
- + Archive(const string&, const string&, const double&, const string&): *this
- + Archive(const Archive&): *this
- + ~Archive(): void
- + setPath(const string&): *this
- + getPath(): string
- + getData(): ofstream
- + setType(const string&): *this
- + getType(): string
- + setSize(const double&); *this + getSize(); double
- + setDocumentId(const string&): *this
- setDocumentia(const si
- + getDocumentId(); string
- loadData(): *this







Gabe Newell

The easiest way to stop piracy is not by putting antipiracy technology to work. It's by giving those people a service that's better than what they're receiving from the pirates.

References

https://www.mongodb.com/community/licensing

http://mongocxx.org/mongocxx-v3/

https://www.mongodb.com/

https://www.mongodb.com/cloud/atlas

http://nosql-database.org/

https://en.wikipedia.org/wiki/Remote procedure call

https://pt.wikipedia.org/wiki/Protocol Buffers

https://en.wikipedia.org/wiki/Graphical user interface

References

https://www.qt.io/

https://github.com/vurtun/nuklear

https://github.com/ocornut/imgui

http://nanapro.org/en-us/

https://en.wikipedia.org/wiki/C%2B%2B

https://www.mises.org.br/Article.aspx?id=17

https://docs.mongodb.com/manual/data-modeling/

 $\underline{https://stackoverflow.com/questions/22835089/modeling-tool-for-nosql-databases/22835380}$

References

https://www.youtube.com/watch?v= Cufy0gMOBA

https://www.youtube.com/watch?v=tRra4_JmD9Y

http://plasmasturm.org/log/xpath101/

https://stackoverflow.com/questions/1733049/p2p-or-distributed-system-implementation

https://en.wikibooks.org/wiki/The World of Peer-to-Peer (P2P)/Building a P2P System

https://realpython.com/python-web-scraping-practical-introduction/

https://en.wikibooks.org/wiki/C%2B%2B Programming