

HOW TO GET BETTER

FEEDBACK

ABOUT ME

- ▶ Tech Lead Manager @Banksalad
 - ▶ Managing `Assets` team
- ▶ iOS Engineer
- ▶ Accessibility Evangelist
- ▶ Daejeon lover



[@SungdoYoo](#)

OVERVIEW

- ▶ Feedback matters
- ▶ How to get good Feedback
 - ▶ TechSpec
 - ▶ TestCode
- ▶ Q&A

**FEEDBACK
MATTERS**

FEEDBACK MATTERS

- ▶ Important things aren't written in Books
 - ▶ Can you learn Love from books?
 - ▶ Maybe some of it. But not all of it
 - ▶ Then, how can you learn important parts of Love?
 - ▶ Constant Feedback
 - ▶ ...based on mutual respect
 - ▶ Same principle can be applied to Software developments

ONLY HUMAN FEEDBACK CAN TELL YOU

- ▶ What kinds of Code is `good` or `readable` to *Them*
- ▶ What variable names is appropriate for *OUR* conventions
- ▶ What are the *their* expectations on *You*
- ▶ How many layers of abstraction do we need in *CURRENT* circumstances
- ▶ ... *and so on*

INDIVIDUALS HAVE LIMITED PERSPECTIVES

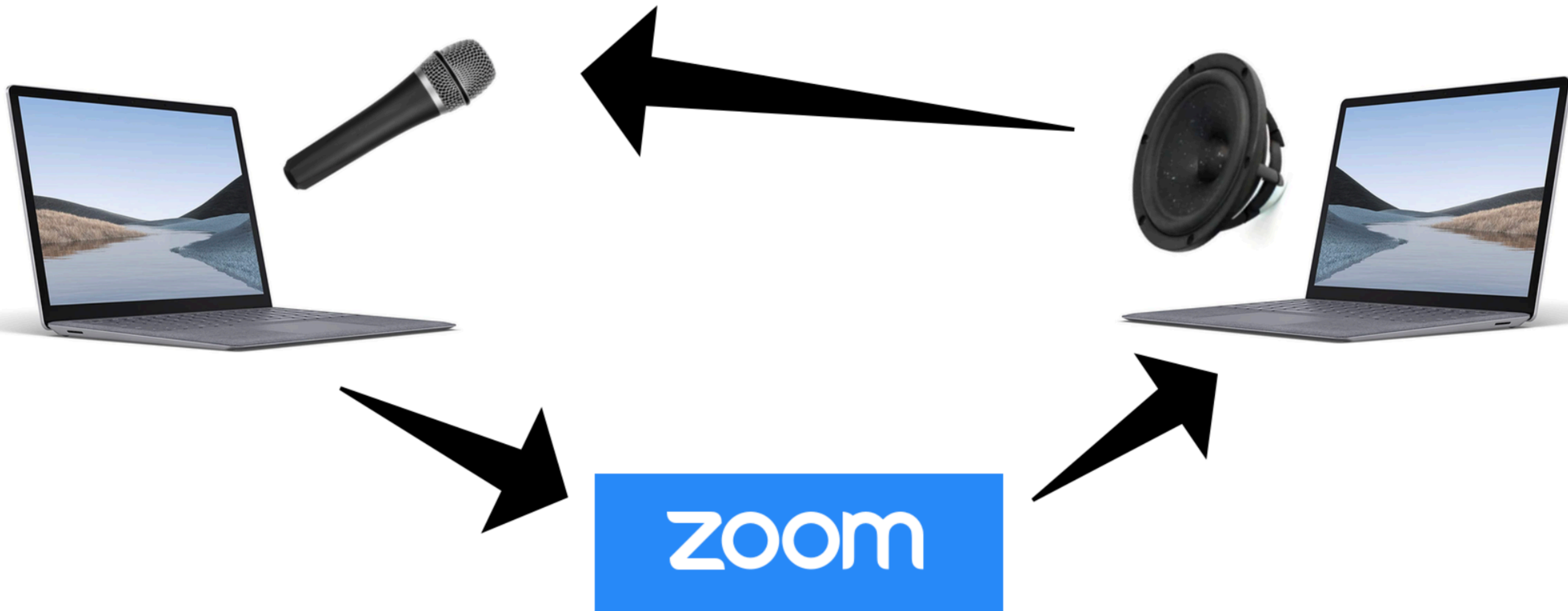
- ▶ Brains relentlessly tries to fool you that everybody sees the world in a way you do.
 - ▶ ...because the world is too complicated for brains to comprehend.
- ▶ Eg1. How can one categorize a person?
 - ▶ Even if category is limited to Age, ethnicity, gender, education, vocation, religion
 - ▶ And even if only there are 4 groups for each category,
 - ▶ combinations are $4^6 == 4,096$
- ▶ Eg2. How should one categorize certain code?
 - ▶ Model? View? Controller? ViewModel?
 - ▶ Domain Layer? Data Layer? Presentation Layer?
 - ▶ DRY vs AHA?
- ▶ Feedback can provide you diverse perspective



INDIVIDUALS HAVE LIMITED KNOWLEDGES

- ▶ Every year, Apple releases more than 200 videos
 - ▶ ... and each video covers multiple new APIs and tools
 - ▶ ... which have multiple combinations of usages and implications
 - ▶ It's not possible to follow every new change.
 - ▶ ... because in industry, unlike in the academic world, "research" and "studying" is not your full-time job.
- ▶ Feedback can provide you with knowledge that you may have missed

FEEDBACK IS COMPOUND INTEREST



IMPORTANT QUESTIONS TO ASK

- ▶ HOW TO GET FEEDBACK

- ▶ WITH BETTER QUALITY

- ▶ WITH BETTER QUANTITY

- ▶ and... some of good mediums we'll cover today is

- ▶ TechSpec

- ▶ TestCode

TECH SPEC

SPEC VS TECH SPEC

- ▶ Spec: When you tap edit button in Asset screen, Edit screen appears
- ▶ TechSpec: If `editButton` on `AssetsViewController` receives `touchUpInside` `UIEvent`, `AssetsViewController` should `push` `EditViewController`
- ▶ SPEC: BankAccount cell should display logo of the bank, name of the product, and balance in the product
- ▶ TechSpec: `BankAccountCell` should should display `organization.logoURL`, `accountDisplayName`, `balane.displayText` declared in `UserBankAccount` model, which is delared in `Bank` module.

WHY?

EVERY SPEC IS MORE COMPLICATED THAN IT LOOKS

- ▶ SPEC: BankAccount cell should display logo of the bank, name of the product, and balance in the product
 - ▶ Where can I get information about BankAccount. Is it already modeled? If so, how and where? Should I model it my self?
 - ▶ If I have to show Logo, which version? circled version? thumbnail version? original version? What are implications of thumbnail version vs original version?
 - ▶ What is `name` of the product? It's original name? or nickname the user declared? How should I handle if the name is too long to fit in one line?
 - ▶ What is `balance`? Evaluated value? Principal? or Withdrawable amount?
 - ▶ How should I handle balance with floating numbers?

IF YOU ESTIMATE RELEASE DATE BASED ONLY ON `SPEC`...

- ▶ It's highly likely that you will make optimistic estimation
- ▶ ...which will result in overwork
- ▶ ...which will result in more bugs
- ▶ ...which will result in even more overwork
- ▶ ...which will delay the release dates, and even impact the business
- ▶ ...which will result in burn-out

INDEXES OF TECHSPEC

- ▶ Background
- ▶ Goal
- ▶ Non-Goal
- ▶ Design
- ▶ Security
- ▶ Test Plan
- ▶ Milestone

뱅크샐러드의 특별한 스펙

테크스펙

Some time
I will be
I will be
I will be
I will be

Some time
I will be

Some time
I will be

Some time
I will be

Some time
I will be

Some time
I will be

BUT

WHAT ABOUT AGILE?

Manifesto for Agile Software Development

We are uncovering better ways of developing software by doing it and helping others do it. Through this work we have come to value:

Individuals and interactions over processes and tools
Working software over comprehensive documentation
Customer collaboration over contract negotiation
Responding to change over following a plan

That is, while there is value in the items on the right, we value the items on the left **more.**

TECHSPEC = CODING IN TEXT

- ▶ If you code in txt, anyone can participate
 - ▶ Without Mac
 - ▶ Without Xcode
 - ▶ Without specific version of ruby or python

문규

김문규
오후 2:14 11월 2일

(p2) custom_net_worth가 의미전달이 더 쉽지 않을까요?

성두

류성두
오후 4:25 11월 2일

이 경우에는 "여러 종류의 자산 상품 중 유저가 선택한 자산만으로 순자산 계산"이라는 의미가 더 강하게 전달되는, 현재의 워딩이 더 적절하다고 생각합니다.

"custom"은 좀 더 넓은 의미같아요. 예컨대 "마이너스 대출을 특정한 방식으로 계산하기" 같은 유저의 custom한 순자산 계산 방식을 적용하기 위한 API인 것으로 헛갈릴 수 있을 것 같아요.

문규

김문규
오후 5:06 11월 2일

ㅎㅎ 표현으로서는 selected net worth에서 selected가 net worth를 꾸며줘서 "선택 된 순자산"이라 의미 전달이 어려울수 있어서 의견 남겼습니다. (원래는 net worth with selected components 가 되어야 하기에...).

아니면 이 옵션 없이 언제나 사용자가 설정한 순자산을 내려줘도 되지 않나요? 문제가 되는 케이스가 있을까요?

성두

류성두
오전 10:13 11월 4일

순자산에 의존하는 다른 서비스 (내자산순위, 연금 계획 등)에 영향을 주지 않기 위한 옵션입니다!

문규 김문규
오후 2:14 11월 2일

(p2) custom_net_worth가 의미전달이 더 쉽지 않을까요?

성두 류성두
오후 4:25 11월 2일

이 경우에는 "여러 종류의 자산 상품 중 유저가 선택한 자산만으로 순자산 계산"이라는 의미가 더 강하게 전달되는, 현재의 워딩이 더 적절하다고 생각합니다.

"custom"은 좀 더 넓은 의미같아요. 예컨대 "마이너스 대출을 특정한 방식으로 계산하기" 같은 유저의 custom한 순자산 계산 방식을 적용하기 위한 API인 것으로 헛갈릴 수 있을 것 같아요.

문규 김문규
오후 5:06 11월 2일

ㅎㅎ 표현으로서는 selected net worth에서 selected가 net worth를 꾸며줘서 "선택 된 순자산"이라 의미 전달이 어려울수 있어서 의견 남겼습니다. (원래는 net worth with selected components 가 되어야 하기에...).

아니면 이 옵션 없이 언제나 사용자가 설정한 순자산을 내려줘도 되지 않나요? 문제가 되는 케이스가 있을까요?


성두 류성두
오전 10:13 11월 4일

순자산에 의존하는 다른 서비스 (내자산순위, 연금 계획 등)에 영향을 주지 않기 위한 옵션입니다!

선택된 텍스트 | rpc ListUserMonthlyTotalAs... ▾

문규 김문규
오후 2:21 11월 2일

이건 분석계 활용하는 것도 고려해보면 좋을 것 같습니다. cc @wonhee@rainist.com

 손원희
오후 2:29 11월 2일

넵 가능성 검토 업무가 계획되어 있습니다!

성두 류성두
오전 10:52 11월 7일

완료됨으로 표시됨

성두 댓글을 추가하면 토론이 재개됩니다...

선택된 텍스트 | 뱅샐머니의 경우 0원이면 자산이... ▾

문규 김문규
오후 2:13 11월 2일

이 부분은 서버에서 구현되나요?

성두 류성두
오후 3:03 11월 2일

넵 맞습니다!

<https://www.notion.so/banksalad/fb635222a7ca416d997e96f85442cb8a?p=e4f35029299343ad865f223765b90bf3&pm=s>

문규 김문규
오후 2:14 11월 2일
(p2) custom_net_worth가 의미전달이 더 쉽지 않을까요?

성두 류성두
오후 4:25 11월 2일
이 경우에는 "여러 종류의 자산 상품 중 유저가 선택한 자산만으로 순자산 계산"이라는 의미가 더 강하게 전달되는, 현재의 워딩이 더 적절하다고 생각합니다.

"custom"은 좀 더 넓은 의미같아요. 예컨대 "마이너스 대출을 특정한 방식으로 계산하기" 같은 유저의 custom한 순자산 계산 방식을 적용하기 위한 API인 것으로 헛갈릴 수 있을 것 같아요.


문규 김문규
오후 5:06 11월 2일
ㅎㅎ 표현으로서는 selected net worth에서 selected가 net worth를 꾸며줘서 "선택 된 순자산"이라 의미 전달이 어려울수 있어서 의견 남겼습니다. (원래는 net worth with selected components 가 되어야 하기에...).

아니면 이 옵션 없이 언제나 사용자가 설정한 순자산을 내려줘도 되지 않나요? 문제가 되는 케이스가 있을까요?

성두 류성두
오전 10:13 11월 4일
순자산에 의존하는 다른 서비스 (내자산순위, 연금 계획 등)에 영향을 주지 않기 위한 옵션입니다!

선택된 텍스트 | rpc ListUserMonthlyTotalAs... ▾

문규 김문규
오후 2:21 11월 2일
이건 분석계 활용하는 것도 고려해보면 좋을 것 같습니다. cc @wonhee@rainist.com

 손원희
오후 2:29 11월 2일
넵 가능성 검토 업무가 계획되어 있습니다!


성두 류성두
오전 10:52 11월 7일
완료됨으로 표시됨

성두 댓글을 추가하면 토론이 재개됩니다...

선택된 텍스트 | 뱅샐머니의 경우 0원이면 자산이... ▾

문규 김문규
오후 2:13 11월 2일
이 부분은 서버에서 구현되나요?

성두 류성두
오후 3:03 11월 2일
넵 맞습니다!
<https://www.notion.so/banksalad/fb635222a7ca416d997e96f85442cb8a?p=e4f35029299343ad865f223765b90bf3&pm=s>

 한지혜
오후 7:08 10월 31일
비율 값은 서버에서 아예 2f로 내려주고 클라에서 일의 자리까지 보여줄 지 소수점까지 보여줄 지 결정할거라 생각했습니당
@sungdoo.yoo@rainist.com
@donghyuk@rainist.com
@seungwoo@rainist.com
<https://github.com/banksalad/idl/pull/4521>

성두 류성두
오후 7:25 10월 31일
저도 그렇게 생각했어요!

다만 1% 이상 자산들의 경우, 서버에서 자체적으로 반올림 해서 소수점이 제거된 값으로 내려올 것으로 기대합니다!

성두 류성두
오후 7:40 10월 31일
지혜님이랑 구두로 논의한 내용
* 요 값은 홈탭에서도 사용하는 값이어서, 홈탭에서도 안정적으로 사용하려면, 우리의 별도 로직이 적용되어서는 안 됨.
* 따라서 여기서의 비율은 소수점 2자리까지 서버에서 정직하게 계산해야 함
* 이렇게 받은 비율 값을 어떤 단위로 반올림 할지는 클라이언트에 넣어야 함

* 이를 위한 추가 클라이언트 테스트케이스 필요

문규 김문규
오후 2:14 11월 2일
(p2) custom_net_worth가 의미전달이 더 쉽지 않을까요?

성두 류성두
오후 4:25 11월 2일
이 경우에는 "여러 종류의 자산 상품 중 유저가 선택한 자산만으로 순자산 계산"이라는 의미가 더 강하게 전달되는, 현재의 워딩이 더 적절하다고 생각합니다.

"custom"은 좀 더 넓은 의미같아요. 예컨대 "마이너스 대출을 특정한 방식으로 계산하기" 같은 유저의 custom한 순자산 계산 방식을 적용하기 위한 API인 것으로 헛갈릴 수 있을 것 같아요.


문규 김문규
오후 5:06 11월 2일
ㅎㅎ 표현으로서는 selected net worth에서 selected가 net worth를 꾸며줘서 "선택 된 순자산"이라 의미 전달이 어려울수 있어서 의견 남겼습니다. (원래는 net worth with selected components 가 되어야 하기에...).

아니면 이 옵션 없이 언제나 사용자가 설정한 순자산을 내려줘도 되지 않나요? 문제가 되는 케이스가 있을까요?

성두 류성두
오전 10:13 11월 4일
순자산에 의존하는 다른 서비스 (내자산순위, 연금 계획 등)에 영향을 주지 않기 위한 옵션입니다!

선택된 텍스트 | rpc ListUserMonthlyTotalAs...

문규 김문규
오후 2:21 11월 2일
이건 분석계 활용하는 것도 고려해보면 좋을 것 같습니다. cc @wonhee@rainist.com

 **손원희**
오후 2:29 11월 2일
넵 가능성 검토 업무가 계획되어 있습니다!

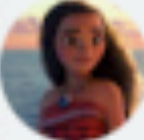
성두 류성두
오전 10:52 11월 7일
완료됨으로 표시됨

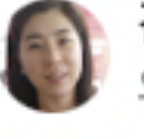
성두 댓글을 추가하면 토론이 재개됩니다...

선택된 텍스트 | 뱅샐머니의 경우 0원이면 자산이...

문규 김문규
오후 2:13 11월 2일
이 부분은 서버에서 구현되나요?

성두 류성두
오후 3:03 11월 2일
넵 맞습니다!
<https://www.notion.so/banksalad/fb635222a7ca416d997e96f85442cb8a?p=e4f35029299343ad865f223765b90bf3&pm=s>

 **한지혜**
오후 7:08 10월 31일
비율 값은 서버에서 아예 2f로 내려주고 클라에서 일의 자리까지 보여줄 지 소수점까지 보여줄 지 결정할거라 생각했습니당
@sungdoo.yoo@rainist.com
@donghyuk@rainist.com
@seungwoo@rainist.com
<https://github.com/banksalad/idl/pull/452>
1

 **김태희**
오후 5:27 11월 9일
추가: ', 1% 이상인 경우 소수점 이하 버림'

성두 류성두
오전 10:48 11월 10일
제안 수용됨

성두 답글을 쓰거나 @로 다른 사용자 추가

성두 오후 7:40 10월 31일
지혜님이랑 구두로 논의한 내용
* 요 값은 홈탭에서도 사용하는 값이어서, 홈탭에서도 안정적으로 사용하려면, 우리의 별도 로직이 적용되어서는 안 됨.
* 따라서 여기서의 비율은 소수점 2자리까지 서버에서 정직하게 계산해야 함
* 이렇게 받은 비율 값을 어떤 단위로 반올림 할지는 클라이언트에 넣어야 함
* 이를 위한 추가 클라이언트 테스트케이스 필요

문규 김문규
오후 2:14 11월 2일
(p2) custom_net_worth가 의미전달이 더 쉽지 않을까요?

성두 류성두
오후 4:25 11월 2일
이 경우에는 "여러 종류의 자산 상품 중 유저가 선택한 자산만으로 순자산 계산"이라는 의미가 더 강하게 전달되는, 현재의 워딩이 더 적절하다고 생각합니다.

"custom"은 좀 더 넓은 의미같아요. 예컨대 "마이너스 대출을 특정한 방식으로 계산하기" 같은 유저의 custom한 순자산 계산 방식을 적용하기 위한 API인 것으로 헛갈릴 수 있을 것 같아요.

문규 김문규
오후 5:06 11월 2일
ㅎㅎ 표현으로서는 selected net worth에서 selected가 net worth를 꾸며줘서 "선택 된 순자산"이라 의미 전달이 어려울수 있어서 의견 남겼습니다. (원래는 net worth with selected components 가 되어야 하기에...).

아니면 이 옵션 없이 언제나 사용자가 설정한 순자산을 내려줘도 되지 않나요? 문제가 되는 케이스가 있을까요?

성두 류성두
오전 10:13 11월 4일
순자산에 의존하는 다른 서비스 (내자산순위, 연금 계획 등)에 영향을 주지 않기 위한 옵션입니다!

선택된 텍스트 | rpc ListUserMonthlyTotalAs...

문규 김문규
오후 2:21 11월 2일
이건 분석계 활용하는 것도 고려해보면 좋을 것 같습니다. cc @wonhee@rainist.com

손원희
오후 2:29 11월 2일
넵 가능성 검토 업무가 계획되어 있습니다
선택된 텍스트 | 연결이 끊겼습니다. 다시 시도해 주세요.

성두 김성민
오전 11:08 11월 9일
ux writing 반영된 문구입니다

성두 류성두
오전 11:57 11월 9일
그리고 이것 Dialog가 아니라토스트로 보여주면 되지요?

선택된

문규 김성민
오후 12:50 11월 9일
넵 맞습니다 피그마에 추가해두었어음
<https://www.figma.com/file/Awmmlm8mfRrTmtWa6VY7Hr/11.09-%EC%88%98?node-id=1703%3A11641>

성두
오후 3:03 11월 2일
넵 맞습니다!
<https://www.notion.so/banksalad/fb635222a7ca416d997e96f85442cb8a?p=e4f35029299343ad865f223765b90bf3&pm=s>

한지혜
오후 7:08 10월 31일
비율 값은 서버에서 아예 2f로 내려주고 클라에서 일의 자리까지 보여줄 지 소수점까지 보여줄 지 결정할거라 생각했습니당
@sungdoo.yoo@rainist.com
@donghyuk@rainist.com
@seungwoo@rainist.com
<https://github.com/banksalad/idl/pull/452>
1

김태희
오후 5:27 11월 9일
추가: ', 1% 이상인 경우 소수점 이하 버림'

성두 류성두
오전 10:48 11월 10일
제안 수용됨
서 자체적으로 내려올 것

성두 답글을 쓰거나 @로 다른 사용자 추가

성두
오후 7:40 10월 31일
지혜님이랑 구두로 논의한 내용
* 요 값은 홈탭에서도 사용하는 값이어서, 홈탭에서도 안정적으로 사용하려면, 우리의 별도 로직이 적용되어서는 안 됨.
* 따라서 여기서의 비율은 소수점 2자리까지 서버에서 정직하게 계산해야 함
* 이렇게 받은 비율 값을 어떤 단위로 반올림 할지는 클라이언트에 넣어야 함
* 이를 위한 추가 클라이언트 테스트케이스 필요

WHEN TECHSPEC IS DONE

- ▶ Most problems that requires 'creativity' are already solved
 - ▶ Such as "Variable name", or "Interface Design"
 - ▶ ... thus leaving works that are relatively easy to estimate amount of
 - ▶ ... which makes you confident on promising release date
- ▶ But even then, you will face changes of Spec during development..!

**WELCOME CHANGING
REQUIREMENTS,
EVEN LATE IN DEVELOPMENT**

SPEC CHANGES DURING DEVELOPMENT

- ▶ Everyone needs to be aware of Spec changes
 - ▶ Developer need to implement it
 - ▶ QAs need to update test cases
 - ▶ Designers need to update Figma or Sketch
 - ▶ Data Analysts need to update hypothesis
- ▶ Version Control Systems, such as Git solves this kind of problem
- ▶ ... and it's univereal problem, thus almost any Software support this

SPEC CHANGES DURING DEVELOPMENT

▶ Everyone

▶ Develop

▶ QAs n

▶ Design

▶ Data A

비율 소수점 정책

비율: V1.Financialasset.NetWorthComponent.proportion2F / 100.0

금액: V1.Financialasset.NetWorthComponent.totalBalanceKrw0F

- 비율 0%이고, 금액 0인 경우
 - "0%"
- 비율 0%이고, 금액 0이 아닌 경우
 - "0.1% 미만"
- 비율 0.1미만인 경우
 - "0.1% 미만"
- 비율 0.1 이상 1.0미만인 경우
 - ex. "0.9%" (소수점둘째자리를 반올림하고 첫째자리까지표현하고 %를 붙인다)
- 비율이 1.0 이상인 경우
 - ex. 10% (소수점을 올버림하고 정수로 표시한다)

파이차트 정책 (IDL.PieLargeOptions)

옵션

- show_tooltip_text(Bool): true (툴팁에 타이틀 노출 여부)
- show_seperator(Bool): true (Cell간 간격 여부)
- show_display_value(Bool): true (display_value 노출 여부)

데이터

- title(String): 자산 타입 명

성두 류성두
오후 11:07 오늘

바꾸기: '버'을(를) '올'(으)로 바꿈

답글을 쓰거나 @로 다른 사용자 추가

▶ Version Control Systems, such as Git solves this kind of problem

▶ ... and it's universal problem, thus almost any Software support this

SPEC CHANGES DURING DEVELOPMENT

- ▶ Everyone
- ▶ Develop
- ▶ QAs n
- ▶ Design
- ▶ Data A
- ▶ Version Control Syst
- ▶ ... and it's universal p

비율 소수점 정책
비율: V1.Financialasset
금액: V1.Financialasset
1. 비율 0%이고, 금액
a. "0%"
2. 비율 0%이고, 금액
a. "0.1% 미만"
3. 비율 0.1미만인 경우
a. "0.1% 미만"
4. 비율 0.1 이상 1.0미
a. ex. "0.9%"
5. 비율이 1.0 이상인 경우
a. ex. 10% (-)
파이차트 정책 (IDL.PieLarg
옵션
- show_tooltip_text
- show_seperator
- show_display_val
데이터
- title(String): 자산

Version history

Current version

2:58 PM
성민 김

컴퍼넌트 이름 제안 merged
컴퍼넌트의 이름을 EditAssetListItem
으로 바꾸고, 앞으로 이 단위에 대해 해당
이름으로 소통하면 좋을 것 같습니다!
성민 김
8 hours ago

2 autosave versions ▶

기본 순서 정책 삭제 merged
기본 정렬 순서는, "자산 탭"의 정책인 것
같아요. 순서편집 화면에서는 자산탭에
보이고 있는 순서를 그대로 반영하게 되
는 거니까.. 여기에 따로 정책을 표시할
Show more
성민 김
8 hours ago

3 autosave versions ▶

주요 정책을 한 곳에 merged
주요 정책들이 한 곳에 모여 있으면 스펙

0

Branches

Active	Yours	Archived	
			+
			...
			...
			...

컴퍼넌트 이름 제안 Merged
성민 김 · Merged 8 hours ago

기본 순서 정책 삭제 Merged
성민 김 · Merged 8 hours ago

주요 정책을 한 곳에 Merged
성민 김 · Merged 8 hours ago

st any Software support this

SUMMARY

- ▶ You can get feedback more easily, from more diverse audience, through Text
 - ▶ Variable names, Interface Design, conventions
 - ▶ TestCases
- ▶ You can make good milestone estimation by finding hidden complexity while writing TechSpec
- ▶ Documents where any team member can access is perfect place where we can keep history of changes of spec.

TEST CODE

WHY WRITE TESTCODE?

- ▶ To... Test Code?
 - ▶ Yes, but that's not the only reason
- ▶ It makes easy to get feedback from
 - ▶ Myself
 - ▶ Computer

FEEDBACK FROM MYSELF

- ▶ Swift API Design Guideline says..
 - ▶ *Clarity at the point of use is your most important goal.*
- ▶ Writing testcode is writing code `at the point of use`.
 - ▶ It literally changes your "point of view", thus making you to give feedback from "other perspective"

TDD ON PAPER

- ▶ You don't need fancy testing framework to practice TDD
 - ▶ You may start with paper or TechSpec
 - ▶ In there, you can design how your code should look in "call sight"
 - ▶ And get feedbacks on TestCases from QA engineers or Designers
- ▶ If you've gotten used to writing TestCases, than you may move on to using fancy Testing Frameworks
 - ▶ ...which may require some overhead to learn and adapt

FEEDBACK FROM COMPUTER

- ▶ You have to utilize computers to generate good feedback.
- ▶ Thousands of TestCases running everytime you commit changes, can give you great feedback on what could go wrong with the change.
 - ▶ If TechSpec or TestCases in Papers are about making better "quality" of feedback,
 - ▶ Automated Tests is about making better "quantity" of feedback

FEEDBACK FROM COMPUTERS

- ▶ Only computer can tell you INVISIBLE implications of a certain change
 - ▶ Logging
 - ▶ Accessibility
- ▶ Only computer can test various combinations of
 - ▶ ScreenSizes (iPhone Mini, iPad, Apple Watch)
 - ▶ Environment (Darkmode, Focus Mode)
 - ▶ system settings (VoiceOver, Larger Fonts, Reduced Motion and so on)

SUMMARY

- ▶ Testcode is not all about Testing
- ▶ It's about having different perspective.
- ▶ It's about getting feedback from your self with different perspective
- ▶ It's about getting feedback from computer
 - ▶ ... in a way that only computers can

LINKS

- ▶ [뱅크샐러드의 특별한 스펙, '테크스펙'](#)
- ▶ [테크스펙으로 모두가 함께 성장하기](#)
- ▶ [Testing in Banksalad](#)
- ▶ [누구나 당장 시작하는 TDD](#)
- ▶ [당신이 TDD를 시도했다가 포기해봤다면](#)