How typescript improves Developer Experience

Kacper Szewczyk

- JS developer at RST Software
- Wrocław / Poland
- Next.js / React Native
- Ex-IT project manager


Microsoft

C# & JavaScript

Script#

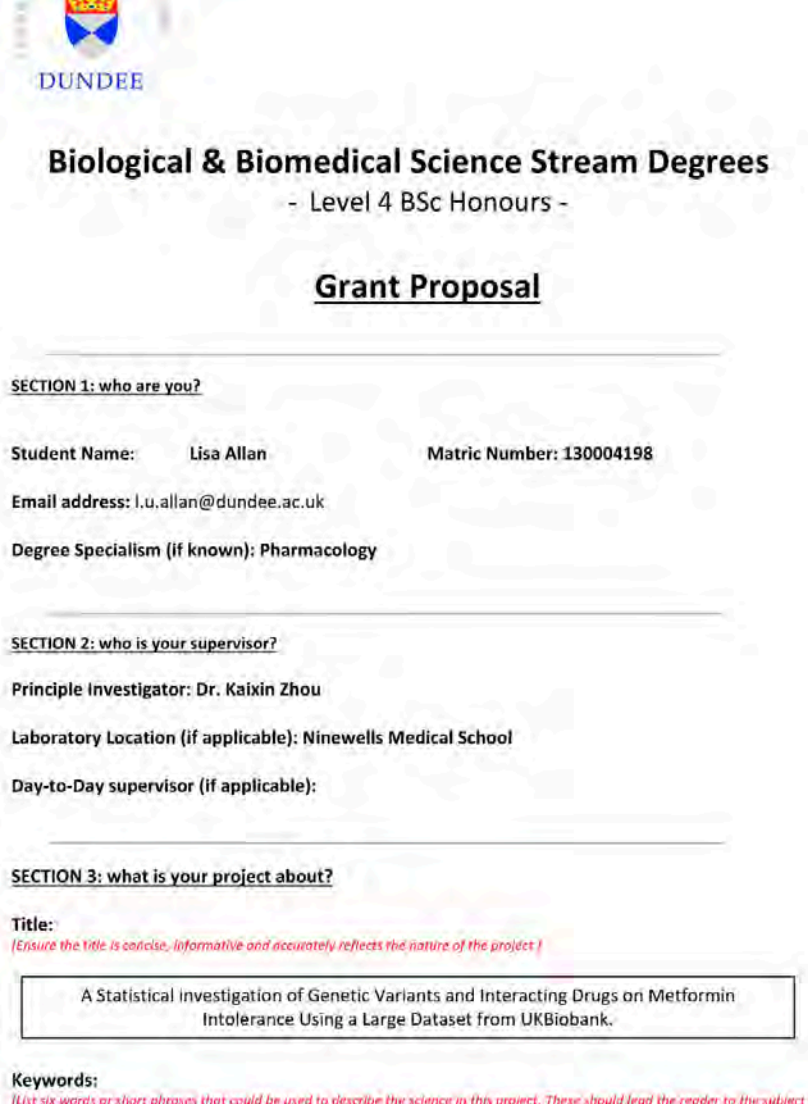
Problems?

TypeScript


Microsoft

Grant

- applicant data
- application number
- information about project
- attachments
- signatures / approvals



DUNDEE

Biological & Biomedical Science Stream Degrees

- Level 4 BSc Honours -

Grant Proposal

SECTION 1: who are you?

Student Name: Lisa Allan **Matric Number: 130004198**

Email address: l.u.allan@dundee.ac.uk

Degree Specialism (if known): Pharmacology

SECTION 2: who is your supervisor?

Principle Investigator: Dr. Kaixin Zhou

Laboratory Location (if applicable): Ninewells Medical School

Day-to-Day supervisor (if applicable):

SECTION 3: what is your project about?

Title:

(Ensure the title is concise, informative and accurately reflects the nature of the project.)

A Statistical investigation of Genetic Variants and Interacting Drugs on Metformin Intolerance Using a Large Dataset from UKBiobank.

Keywords:

(List six words or short phrases that could be used to describe the science in this project. These should lead the reader to the subject

Applicant data

```
type Address = {
    city: string;
    postalCode: string;
    country: string;
}
export type Applicant = {
    firstName: string;
    lastName: string;
    email: string;
    phoneNumber: string;
    address: Address;
    SSN: string;
}
```

```
export type Applicant = {
    firstName: string;
    lastName: string;
    email: string;
    phoneNumber: string;
    address: Address;
    SSN?: string;
    birthDate?: Date;
}
```

```
export type Applicant = {
    firstName: string;
    lastName: string;
    email: string;
    phoneNumber: string;
    address: Address;
    SSN?: string;
    birthDate?: Date;
    companyName?: string;
    taxId?: string;
}
```

Applicant data

```ts
export type PrivateApplicant = {
    firstName: string;
    lastName: string;
    email: string;
    phoneNumber: string;
    address: Address;
    SSN: string;
}
```

```ts
export type PrivateApplicantWithoutSSN = {
    firstName: string;
    lastName: string;
    email: string;
    phoneNumber: string;
    address: Address;
    birthDate: string;
}
```

```ts
export type CompanyApplicant = {
    firstName: string;
    lastName: string;
    email: string;
    phoneNumber: string;
    address: Address;
    companyName: string;
    taxId: string;
}
```

```ts
export type Applicant =
  PrivateApplicant |
  PrivateApplicantWithoutSSN |
  CompanyApplicant;
```

Applicant data

```
type BaseApplicant = {
    firstName: string;
    lastName: string;
    email: string;
    phoneNumber: string;
    address: Address;
}
```

```
export type PrivateApplicant = BaseApplicant & {
    SSN: string;
}
```

```
export type PrivateApplicantWithoutSSN = BaseApplicant & {
    birthDate: string;
}
```

```
export type CompanyApplicant = BaseApplicant & {
    companyName: string;
    taxId: string;
}
```

Applicant data

```
type Address = {
    city: string;
    postalCode: string;
    country: string;
}

export type PrivateApplicant = {
    firstName: string;
    lastName: string;
    email: string;
    phoneNumber: string;
    address: Address;
    SSN: string;
}
```

```
export type PrivateApplicantWithoutSSN =
  Omit<PrivateApplicant, "SSN"> & {
    birthDate: Date;
};

export type CompanyApplicant = Pick<PrivateApplicant,
  "firstName" | "lastName" |
  "email" | "phoneNumber" |
  "address"> & {
    companyName: string;
    taxId: string;
}
```

Application number

```
export type ApplicationNumber = string
// AB123456/2021

type ApplicationTypeCodes = "AB" | "CD" | "EF" | "GH" | "IJ" | "KL" | "MN" | "OP" | "QR" | "ST" | "UV" | "WX" | "YZ";

type ApplicationYear = '2020' | '2021' | '2022' | '2023' | '2024'

export type ApplicationNumber = `${ApplicationTypeCodes}${number}/${ApplicationYear}`
```

Application number

```
type ApplicationTypeCodes = "AB" | "CD" | "EF" | "GH" | "IJ" | "KL" | "MN" | "OP" | "QR" | "ST" | "UV" | "WX" | "YZ";

type ApplicationYear = '2020' | '2021' | '2022' | '2023' | '2024'

export type ApplicationNumber = `${ApplicationTypeCodes}${number}/${ApplicationYear}`



const appNumber: ApplicationNumber = 'AB202130912309123/2020'
```

```ts
type ComputeRange<
  N extends number,
  Result extends Array<unknown> = []
> = Result['length'] extends N
  ? Result
  : ComputeRange<N, [...Result, Result['length']]>;

type ApplicationTypeCodes = "AB" | "CD" | "EF" | "GH" | "IJ" | "KL" | "MN" | "OP" | "QR" | "ST" | "UV" | "WX" | "YZ";

type YearEnding = `${ComputeRange<5>[number]}`

type ApplicationYear = `202${YearEnding}`

export type ApplicationNumber = `${ApplicationTypeCodes}${number}/${ApplicationYear}`
```

Attachments

```ts
import { ComputeRange } from "../utils/computeRange";

type AttachmentType = "EXPENSES" | "INCOME" | "OTHER";

export type AttachmentDetailsPayload = {
    description: string;
    type: AttachmentType;
    pageCount: number;
    file: File;
}

export type AttachmentDetails = Pick<AttachmentDetailsPayload, "description" | "type" | "pageCount"> & {
    fileUrl: string;
}

type AttachmentName = `Attachment ${ComputeRange<5>[number]}`

export type AttachmentList = Record<AttachmentName, AttachmentDetails>;
```

Attachments

```ts
import { ComputeRange } from "../utils/computeRange";

type AttachmentType = "EXPENSES" | "INCOME" | "OTHER";

export type AttachmentDetailsPayload = {
    description: string;
    type: AttachmentType;
    pageCount: number;
    file: File;
}

export type AttachmentDetails = Pick<AttachmentDetailsPayload, "description" | "type" | "pageCount"> & {
    fileUrl: string;
}

type AttachmentName = `Attachment ${ComputeRange<5>[number]}`

export type AttachmentList = Record<AttachmentName, AttachmentDetails>;
```

Application

```
export type Application = {
   applicant: Applicant;
   applicationDate: Date;
   applicationNumber: ApplicationNumber;
   attachments: Partial<AttachmentList>;
}
```

Type guards

```
export function isObjectWithKey(
  data: unknown,
): data is { [key: string]: unknown } {
  return typeof data === 'object' && data !== null;
}

export const isPrivateApplicant = (applicant: unknown): applicant is PrivateApplicant => {
   return (
    isObjectWithKey(applicant) &&
    applicant.hasOwnProperty('SSN')
  );
}
```

Type guards

```
type PrivateApplicant = { SSN: string; }

type Applicant = PrivateApplicant | string;

function isObjectWithKey(
  data: unknown,
): data is { [key: string]: unknown } {
  return typeof data === 'object' && data !== null;
}
const isPrivateApplicant = (applicant: unknown): applicant is PrivateApplicant => {
   return (
    isObjectWithKey(applicant) &&
    applicant.hasOwnProperty('SSN')
  );
}

const printApplicant = (applicant: Applicant) => {
  if (isPrivateApplicant(applicant)) {
    // ...
    console.log(applicant.SSN);
  }
}
```

Readonly

```
type Application = { applicationNumber: string;}
type ReadonlyApplication = Readonly<Application>;

const application: ReadonlyApplication = {
  applicationNumber: 'AB123455/2021',
};

application.applicationNumber = 'AB123456/2021';
```

Readonly

```
type VerificationStatus = 'ACCEPTED' | 'REJECTED' | 'PENDING';

type ApplicationWithAcceptance = Readonly<Application> & {
  verificationDate: Date;
  verificationStatus: VerificationStatus;
  verifiedBy: string;
};
```

Draft

```
type Applicant = {
    firstName: string;
    lastName: string;
    email: string;
    address: Address;
    SSN: string;
}

type Application = {
   applicant: Applicant;
   applicationDate: Date;
   applicationNumber: 'AB123456/2021' |'AB123456/2022' ;
}

export type Nullable<T> = null | T;
export type DeepNullable<T> = {
  [K in keyof T]: Nullable<DeepNullable<T[K]>>;
};

type ApplicationDraft = DeepNullable<Application>;
```

Key takeways

- User advanced types to create complex types
- Type guards allows for save and fast type checks

Sources

- https://michaelsync.net/2007/10/29/script-c-to-javascript-converter/
- https://www.typescriptlang.org/docs/handbook/typescript-in-5-minutes.html

Kacper Szewczyk

- JS developer at RST Software
- Next.js / React Native
- Ex-IT project manager

Q&A