

DOCUMENTACIÓ MECANISMES D'EXCEPCIONS



IES Enric Valor de Pego
Curs escolar 2023-2024
Desenvolupament d'aplicacions web
2n curs

Professorat participant
Vicent Jordà

Alumne participant
Noah Calafat



IES Enric Valor
C/ Duanes, 17
03780 - Pego
966 409 960
03007613@gva.es



**GENERALITAT
VALENCIANA**
Conselleria d'Educació,
Cultura i Esport



Fons Social Europeu

L'FSE inverteix en el teu futur

UNIÓ EUROPEA

Índex

Introducció	2
Què és una jerarquia d'excepcions?	3
Què és una excepció?	3
Quines són les classes d'excepcions més usades?	4
Diferències i quan es deuen d'usar:	5
Conclusió:	6
Webgrafia	7



Introducció

Aquest document té com a objectiu proporcionar una comprensió detallada de la jerarquia de les excepcions a PHP. Entendre aquesta jerarquia és fonamental per a la gestió d'errors i excepcions en les aplicacions PHP, permetent una millor depuració i manteniment del codi.

Explorarem les classes d'excepcions predefinides, com la classe base Exception i altres excepcions comunes com RuntimeException, InvalidArgumentException i PDOException. També tractarem la creació d'excepcions personalitzades, que et permeten capturar i gestionar errors específics a la teva aplicació.





Què és una jerarquia d'excepcions?

Les jerarquies d'excepcions a PHP, o també conegudes com a "jerarquia d'errors," es refereixen a l'estructura organitzada de les classes d'excepcions en el llenguatge de programació PHP. Aquesta jerarquia proporciona una manera de gestionar i capturar errors i excepcions de manera més estructurada i eficient.

Què és una excepció?

Les excepcions són esdeveniments inesperats o errors que poden ocórrer durant l'execució d'un programa i que poden ser gestionats per l'aplicació mitjançant el codi.

En PHP, les excepcions estan organitzades com una jerarquia de classes, amb una classe base comuna anomenada "Exception." Aquesta classe base és estesa per altres classes d'excepcions més específiques, les quals permeten als desenvolupadors capturar i gestionar diferents tipus d'errors d'una manera més precisa.



Arbre jerarquia d'excepcions

La classe Exception de PHP és la classe base per a totes les excepcions personalitzades en PHP.

getMessage(): Retorna el missatge de l'excepció com una cadena de caràcters.

getCode(): Retorna el codi d'error de l'excepció com un enter.

getFile(): Retorna la ruta al fitxer en el qual es va llençar l'excepció com una cadena de caràcters.

getLine(): Retorna el número de línia en la qual es va llençar l'excepció com un enter.

getTrace(): Retorna una matriu de la pila de seguiment de l'excepció. Cada element de la matriu és un array associatiu amb informació sobre la crida a la funció que va provocar l'excepció.

getTraceAsString(): Retorna la pila de seguiment de l'excepció com una cadena de caràcters.

getPrevious(): Retorna l'excepció anterior, si aquesta excepció va ser llançada a causa d'una altra excepció. Això permet fer un seguiment de la causa original de l'excepció.

__toString(): Retorna una representació en cadena de l'excepció que inclou el missatge, el codi, el fitxer i la línia on va ocórrer l'excepció.

__construct(): Constructor per crear una instància de l'excepció. Pots proporcionar un missatge i un codi d'error com a arguments opcionals.

Aquests mètodes permeten accedir a la informació sobre l'excepció, com ara el seu missatge, el seu origen en el codi, la seva pila de seguiment i més. Això és útil per depurar i registrar errors de manera efectiva a les nostres aplicacions.



Quines són les classes d'excepcions més usades? Exemples

La jerarquia d'excepcions en PHP inclou diverses classes d'excepcions predefinides que estenen la classe Exception. Algunes de les classes d'excepcions més comunes inclouen:

1. **RuntimeException:** S'utilitza per a errors en temps d'execució que no poden ser detectats prèviament.

```
try {  
    //Això generaria una excepció RuntimeException  
}  
catch (RuntimeException $e) {  
    echo "Error d'argument no vàlid: " . $e->getMessage();  
}
```

2. **InvalidArgumentException:** S'utilitza per a errors relacionats amb arguments no vàlids passats a una funció o mètode.

```
try {  
    //Això generaria una excepció InvalidArgumentException  
}  
catch (InvalidArgumentException $e) {  
    echo "Error d'argument no vàlid: " . $e->getMessage();  
}
```

3. **PDOException:** Específica per a errors relacionats amb bases de dades i connexions de bases de dades.

```
try {  
    //Això generaria una excepció PDOException  
}  
catch (PDOException $e) {  
    echo "Error d'argument no vàlid: " . $e->getMessage();}
```



Diferències i quan es deuen d'usar:

És important entendre les diferències entre les tres excepcions: RuntimeException, InvalidArgumentException i PDOException, i quan utilitzar cadascuna depèn del tipus d'error o situació que estàs tractant.

RuntimeException:

- **Què representa:** S'utilitzarà per a errors en temps d'execució que no poden ser detectats prèviament, com ara divisió per zero o crides a mètodes en objectes que no existeixen.
- **Quan utilitzar-la:** S'utilitzarà RuntimeException quan estigues tractant un error que és inesperat i no es pot gestionar prèviament. És una excepció genèrica que es pot utilitzar per a situacions imprevistes.

InvalidArgumentException:

- **Què representa:** S'utilitzarà per a errors relacionats amb arguments invàlids passats a una funció o mètode.
- **Quan utilitzar-la:** S'utilitzarà InvalidArgumentException quan detectem que els arguments passats a una funció o mètode no són vàlids. Aquesta excepció ajuda a identificar problemes específics amb les dades d'entrada.

PDOException:

- **Què representa:** Especifica per a errors relacionats amb bases de dades i connexions de bases de dades. Aquesta excepció és comuna quan es treballa amb PDO (PHP Data Objects) per gestionar connexions i consultes de bases de dades.
- **Quan utilitzar-la:** S'utilitzarà PDOException quan es produeixen problemes en l'accés a una base de dades o en l'execució de consultes SQL. Aquesta excepció està dissenyada específicament per a errors relacionats amb bases de dades.



Conclusió:

Hem explorat les classes d'excepcions predefinides com la classe base Exception i altres excepcions comunes com RuntimeException, InvalidArgumentException i PDOException. A més, hem examinat com crear i utilitzar excepcions personalitzades per a capturar errors específics a la nostra aplicació.

En resum, una gestió d'excepcions adequada i una comprensió sòlida de la jerarquia d'excepcions són essencials per a escriure codi PHP robust i mantenible. Utilitzar les excepcions correctes en les situacions apropiades millorarà la fiabilitat i la seguretat de les aplicacions PHP, permetent-nos oferir una millor experiència als usuaris.



IES Enric Valor
C/ Duanes, 17
03780 - Pego
966 409 960
03007613@gva.es



**GENERALITAT
VALENCIANA**
Conselleria d'Educació,
Cultura i Esport



Fons Social Europeu

L'FSE inverteix en el teu futur

UNIÓ EUROPEA

Webgrafia

- [YouTube](#) -> [Consulta: 29 octubre 2023 | 19:00]
- [PHP](#) -> [Consulta: 29 octubre 2023 | 19:00]

WEBTD - YouTube [en línia] <<https://www.youtube.com/@webtd97>> [Consulta: 29 octubre 2023]

PHP Documentation [en línia] <<https://www.php.net/docs.php>> [Consulta: 29 octubre 2023]