

# FastJson的范畴和选项

该文档主要记录FastJson的范畴和选项

范畴	选项
1:float	1-1:不存在；1-2:存在
2:double	2-1:不存在；2-2:存在
3:short	3-1:不存在；3-2:存在
4:byte	4-1:不存在；4-2:存在
5:int	5-1:不存在；5-2:存在
6:long	6-1:不存在；6-2:存在
7:boolean	7-1: 不存在；7-2: 存在
8:char	8-1:不存在；8-2:存在
9:date	9-1:不存在；9-2:存在
10:string	10-1: 不存在；10-2: 存在
11:enum	11-1: 不存在；11-2: 存在
12: Map	12-1: 不存在；12-2: 存在
13: List	13-1: 不存在；13-2: 存在
14: Set	14-1: 不存在；14-2: 存在

（注：date范畴若9-2选项存在时，则从yyyy-MM-dd HH:mm:ss；10-3:yyyy/MM/dd HH:mm:ss；10-4:yyyy年M月d日 HH:mm:ss；10-5:yyyy年M月d日 H时m分s秒；10-6:yyyy년M월d일 HH:mm:ss；10-7:MM/dd/yyyy HH:mm:ss；10-8:dd/MM/yyyy HH:mm:ss；10-9:dd.MM.yyyy HH:mm:ss；10-10:dd-MM-yyyy HH:mm:ss；10-11:yyyyMMdd；10-12:yyyy/MM/dd；10-13:yyyy年M月d日；10-14:yyyy년M월d일；10-15:MM/dd/yyyy；10-16:dd/MM/yyyy；10-17:dd.MM.yyyy；10-18:dd-MM-yyyy中任选一种。）

## 获取范畴的方法

由于FastJson缺少规格说明书，我们主要通过分析FastJson的历史故障信息以及FastJson的benchmark中存在的测试用例，得到范畴和选项。序列化和反序列化的主要目的是保存程序在某个时刻的状态信息，即程序中成员变量的值以及方法参数的值。在识别范畴的过程中，我们考虑如下2个方面：（1）在对FastJson历史故障进行分析时，我们发现大多数反序列化故障是由于基本数据类型在特定情况下引起的；（2）FastJson的benchmark中的测试用例仅将多个基本数据类型作为成员变量构成JB。